FACULTY OF ENGINEERING
AND ARCHITECTURE

Photonics-Based Machine Learning to Speed up and Simplify Label-Free Flow Cytometry

Alessio Lugnan

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Photonics Engineering

Supervisors
Prof. Peter Bienstman, PhD* - Prof. Joni Dambre, PhD**

* Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

** Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

September 2021

GHENT
UNIVERSITY

ISBN 978-94-6355-513-5
NUR 926, 959
Wettelijk depot: D/2021/10.500/61

Members of the Examination Board

Chair

Prof. Patrick De Baets, PhD, Ghent University

Other members entitled to vote

Prof. Adonis Bogris, PhD, University of West Attica, Greece
Prof. Nicolas Le Thomas, PhD, Ghent University
Prof. Filip Strubbe, PhD, Ghent University
Prof. Chao Wang, PhD, University of Kent, United Kingdom

Supervisors

Prof. Peter Bienstman, PhD, Ghent University
Prof. Joni Dambre, PhD, Ghent University

Acknowledgments

Without doubt, it is to Peter Bienstman that goes the most important acknowledgment regarding my PhD, but now I will follow another kind of order. So I first thank my family, who supported my studies and gave me the great privilege of doing what I wanted to do. My father Livio and grandfather Gianfranco were the ones that started this flame of passion for discovery, curiosity and thinking for the sake of thinking, that I still feel burning strong. They, and my mother Silvia most of all, together with my grandmother Daria, and zia Gina and many other relatives, allowed me to and helped me to keep this flame alive for a long time. With their trust and pride, and with their affection. And thanks to my brother Gabriele, for being a grown up Gabriele.

Then I would like to thank all my teachers, but especially those who tried their best to make us fragile pupils better students and persons. I was lucky because I had several of this kind of teachers, and I still have. I will name some, to remark that I have not forgotten them and that for me their work really mattered: the very special duo maestre Chiara e Carla, at primary school; prof. Caproni, who ignited my passion for poetry and writing, and who believed in me, and prof. Baviera and Jeronimo at middle school, and prof. Zeni who lent his delicate musical ear to our terrible and piercing flute technique, and prof. Perotti, who showed us the wild side of art; at high school prof. Salvetti and prof. Muselli, who tried hard to be good teachers in their very different and personal ways, and extremely unpredictable prof. Setti, who could let his passion towards art history be transferred to us in a mysterious but effective way. At the University of Trento, prof. Giovanni Prodi, who worked hard to teach us rigorous experiment designing and results analysis during the first year. Dr. Fernando Ramiro Manzano, who showed me the power of true interactivity during lessons. Dr. Mattia Mancinelli and Prof. Lorenzo Pavesi, who were great supervisors of my Master's thesis work, through which I entered the exciting world of neuromorphic computing with photonics. And many others, comprising all my classmates who have been friends and key elements of my path as a student. Of course I also thank Italy and Italians, for being able to access very affordable and high-quality education.

A very big role in making my life in Ghent enjoyable, and my work at the Photonics Reasearch Group a very positive experience, was played by friends, some of which I really felt have been my family in Belgium. Together we went through the peculiar experience of being PhD students, expats, and of being expats during Covid pandemic. My first friend in Belgium was Omar, when I most needed one, and thanks to him I met other great friends. The closest ones: Alejandro, Mahmoud and Savvina, who could even withstand my long and intense philosophical digressions. We shared many fun moments and memorable adventures, together with other unforgettable colleagues and friends: Nina, Sanja, Ana, and our special and cheerful neighbours: Alexandros and Brianna. And then Fabio, who often discussed about research with me and who could give me very needed advices, Kristof and Hanna, Tommaso and Zaira, Gregory and Zhanna, my superefficient office mate Jeroen, Camiel, Irfan, Clemens and many others, thank you! A special sportive thank you goes to all the colleagues and friends who played ping pong at the 2nd floor after work. It has been perfect to blow off some steam, but it has also been a precious opportunity to get together, have genuine fun and meet new people. We can say we built a sort of special ping pong tradition at iGent, for example with our own unviolable "Dummy"game, to decide who starts serving.

My deepest thank you goes to Margherita, who fills a big hole in my heart and makes every single thing more vivid and alive. From her company I have learned so many things that changed me so much for the better. To stay relevant, without Margherita I could have been a so much worse PhD student, and the PhD path could have been for me much heavier and less enjoyable. She have always tried very hard to support my passion for research and to adapt her life and ambitions to it. She left Italy and jumped into this Belgian adventure with me, and it is only just fair for me to reciprocate her support the best I can. Noticeably, she is unbelievably and genuinely interested in my research, when everyone else in her stead would be extremely bored by listening to the technical detail, the concocted and ever-changing insights into my work.

Now it is time to thank my supervisor, Peter Bienstman. When I first read the announcement of the open PhD position, it was written that the opportunity for "blue sky research"was offered. And "blu sky"gives quite the precise idea of how I felt. It can be terrifying, or it can be beautiful and exciting. For me it was mostly the latter, not only because I like to "fly", but also because I was given a very high-quality aircraft and I have been capably looked after and supported by "ground control". The experience of being a PhD student is overly dependent on how much your supervisor fits you and how much he\she cares about you as a researcher and as a person. I feel very lucky, because Peter has been nearly perfect. From his example and suggestions I learned (and I continuously try to improve) how to be passionate about research, and caring with the people I work with, in a light way (as opposed to heavy). And to be "lazy in a smart way". I find that these modalities can simplify a researcher's life a great deal. Still, I believe I will never be able to write, revise and proofread as efficiently as him, not even close. My second promotor, Joni Dambre, has been a very good match with Peter in helping me and supervising my work. I could learn a lot about machine learning and its application from her didactic and rigorous supervision.

Moreover, I wish to thank the members of the Examination Board, for having read this dissertation and for having improved it with valuable comments, corrections and thought-provoking questions.

Finally, I must acknowledge the exceptional quality of the environment that surrounded me. Starting from Ghent University, the Faculty and Doctoral School, with their inclusiveness and continuous aim for all-round excellence. Most of all, I want to deeply thank all the organizers and staff of the Photonics Research Group, who always provided me with prompt and high-quality support and who enabled me to conduct my research in a smooth way. I felt welcome, valorized and part of a great team. Thanks to their effort and ability I could continuously learn and professionally grow. And another big thank you goes to all my colleagues, because they were always ready to help me and to collaborate for the common good. We had incredible, unbelievable discussions during lunch and coffee breaks (and I think I have already mentioned ping pong). Special thanks go to my colleagues in our reservoir computing subgroup. Bendix and Omar, in different ways, helped me a lot to get started. Floris was always ready to help me, especially with computer and coding issues. Thank you Andrew, Matthias, Stijn and Chonghuai for your assistance and collaboration. I closely collaborated with Jeremy and Emmanuel and I am grateful to them for trying their best to develop our flow cytometry experiment. Finally, I am glad to see that brilliant PhD students, Muhammed and Sarah, have joined the team.

Gent, July 2021
Alessio Lugnan

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

ANN	Artifical Neural Network

B

BC	Beneficial Confusion

C

CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network

D

DB	Decision Boundary
DG	Diffraction Grating

E

ELM	Extreme Learning Machine

F

FDTD	Finite-Difference Time Domain
FFT	Fast Fourier Transform

G

GPU	Graphical Processing Unit

L

LDA	Linear Discriminant Analysis

M

mAP	mean Average Precision
ML	Machine Learning

N

NDG	No Diffraction Grating
NN	Neural Network

P

PCA	Principal Component Analysis
PDF	Probability Density Function
PMMA	Poly(methyl methacrylate)

R

RC	Reservoir Computing

S

SLM	Spatial Light Modulator
SNR	Signal to Noise Ratio

U

UM	Uniform Mislabelling

W

WBC	White Blood Cell

Nederlandse Samenvatting

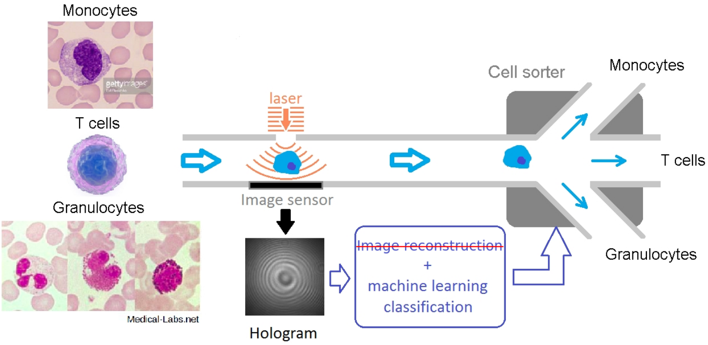
In de afgelopen jaren is machine learning, en in het bijzonder deep learning, toegepast om een breed scala aan problemen aan te pakken, waardoor automatisch taken kunnen worden uitgevoerd die te moeilijk zijn voor traditioneel computergebruik. Populaire en krachtige modellen zoals diepe neurale netwerken brengen echter vaak hoge rekenkosten met zich mee, zowel bij hun training als bij hun toepassing. Door een deel van een machine learning-algoritme in speciale hardware te implementeren, kunnen grote verbeteringen in computerefficiëntie worden bereikt. Desalniettemin is het bijzonder uitdagend om complexe fysieke systemen te fabriceren, b.v. een fysiek kunstmatig neuraal netwerk, dat kan worden afgestemd, gemeten en gemodelleerd in voldoende detail en met voldoende precisie om conventionele machine learning-training mogelijk te maken.

Deze uitdagingen kunnen (althans gedeeltelijk) worden omzeild door willekeurige en niet-lineaire fysieke systemen te combineren die slechts marginale afstembaarheid en observeerbaarheid vereisen, met eenvoudige en computationeel goedkope machine learning-technieken op basis van lineaire bewerkingen, zoals lineaire regressors of classificaties. Een dergelijke benadering wordt gewoonlijk benoemd als een hardwareversie van reservoir computing (voor dynamische hardware met feedbacklussen) of extreme leermachinemethoden (in de afwezigheid van feedbacklussen). Deze technieken hebben recentelijk goede prestaties geleverd in verschillende soorten toepassingen en ze laten een veel eenvoudigere en snellere training toe in vergelijking met hun conventionele tegenhangers in machine learning of deep learning. De rol van het willekeurige niet-lineaire fysieke systeem is om de invoer over te brengen naar een rijkere, hoger dimensionele representatie, om de rekenkracht van het daaropvolgende lineaire machine learning-model aanzienlijk te vergroten.

In dit proefschrift passen we deze aanpak toe door gebruik te maken van de extreme verwerkingssnelheid van fotonische systemen om microscopisch kleine objecten, zoals biologische cellen of microdeeltjes, te classificeren terwijl ze in een microfluïdisch kanaal stromen. Dat wil zeggen, we streven ernaar om machine learning-bewerkingen in labelvrije microflowcytometrie te vereenvoudigen en te versnellen. De computationele kosten van traditionele algoritmen fungeren inderdaad vaak als een bottleneck in dit type applicatie, waardoor de doorvoer van online operaties, zoals celsortering, aanzienlijk wordt beperkt.

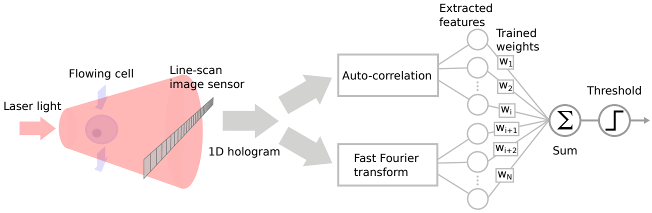
1 Efficiënte classificatie van witte bloedcellen in beeldvormende labelvrije microflowcytometrie

We begonnen dit doctoraatsonderzoek in samenwerking met onderzoekers van imec (een R&D-hub voor nano- en digitale technologieën) die een labelvrije beeldvormende flowcytometer ontwikkelden voor high-throughput sortering van witte bloedcellen, gebaseerd op een lensvrije inline digitale holografische microscopiemethode. In eenvoudigere woorden, ze lieten drie verschillende soorten witte bloedcellen (monocyten, T-cellen en granulocyten) stromen in een transparant microfluïdisch kanaal, en ze registreerden de interferentiepatronen die door deze cellen werden geprojecteerd wanneer ze werden belicht met zichtbaar laserlicht dat door een microscopisch kleine opening scheen (zie Fig. 1). Bij digitale holografische microscopie worden de verworven patronen (hologrammen genoemd) ingevoerd in een computationeel duur beeldreconstructie-algoritme om het celbeeld te verkrijgen. Ons doel was echter om een machine learning-algoritme rechtstreeks toe te passen op het onbewerkte hologram, waardoor de vereiste berekeningen voor celclassificatie aanzienlijk werden verminderd.



Figuur 1: De belangrijkste functionaliteiten van de flowcytometer (ontwikkeld door imec) die de beschouwde hologrammen genereerde. Drie verschillende typen witte bloedcellen stromen in een microfluïdisch kanaal. Zichtbaar laserlicht wordt door een opening loodrecht op het kanaal geschenen en het resulterende diffractiepatroon wordt op een beeldsensor geprojecteerd. De aanwezigheid van een stromende cel, waarvan de lichtabsorptie verwaarloosbaar is, wijzigt het interferentiepatroon dat door de sensor wordt verkregen. Het verkregen beeld, celhologram genaamd, bevat informatie over de 3D-brekingsindexstructuur van de cel, die kan worden gebruikt om het celbeeld te reconstrueren (digitale holografische microscopie). Deze bewerking is rekenkundig duur, daarom hebben we direct een machine learning-classificator toegepast op onbewerkte celhologrammen.

Om de computationele kosten van celclassificatie verder te verlagen en dus de maximale doorvoer van celsortering te verhogen, moesten we de grote dimensionaliteit (gegeven door ongeveer een miljoen pixelwaarden) van de celhologramafbeeldingen drastisch verminderen door minder kenmerken te extraheren en enkel de relevante en niet-overtollige over te houden. We hebben dit gedaan door de originele 2D-patronen om te zetten in 1D-patronen door middel van pixeloptelling langs de kanaalrichting, waarbij we het resultaat benaderen dat zou verkregen worden met behulp van een 1D-beeldsensor. Daarna hebben we geschikte translatie-invariante functies toegepast (autocorrelatie en snelle Fouriertransformatie) en hebben we de resulterende waarden ingevoerd in een lineaire classificator op basis van logistische regressie (Fig. 2). Een dergelijke pijplijn voor het extraheren van kenmerken was bedoeld om de ruis te verminderen die wordt geïntroduceerd door de variabiliteit van de celpatroonpositie t.o.v. het beeldcentrum.



Figuur 2: De voorgestelde classificatiepijplijn. We verkrijgen 1D-hologrammen door 2D-hologrampixels langs de stroomrichting bij elkaar op te tellen, wat een benadering is van het gebruik van een lijnscanbeeldsensor. Geschikte autocorrelatie en snelle Fourier-transformatie waarden worden berekend op basis van het 1D-hologram (kenmerkextractie). Deze kenmerken worden vervolgens gewogen en opgeteld door een lineaire classificator (logistische regressie). Tijdens de training wordt een optimale set gewichten voor elke klasse geleerd. De gewogen som met de hoogste uitkomst geeft de celklasse aan die door de classificator wordt herkend.

Deze computationeel efficiënte classificatiepijplijn vertoonde bevredigende classificatieprestaties wanneer deze werd toegepast op de gegevens die door onze partners werden verstrekt. We ontdekten echter dat de verkregen resultaten sterk vertekend waren door de correlatie tussen langzame afwijkingen in meetomstandigheden (die de laserstraal die de cellen verlichtte beïnvloeden) en de celklassen. In het bijzonder waren hologrammen van elk celtype verkregen in een enkele lange meting, de een na de ander, en daarom waren de gegenereerde gegevens gemarkeerd door de afwijkende meetomstandigheden. Hierdoor kon ons machine learning-algoritme gemakkelijk de beeldclassificatie leren door gebruik te maken van de meetconditie-informatie, in plaats van de feitelijke celinformatie. Een dergelijk machine learning-probleem, shortcut learning of dataset bias genoemd, wordt vaak onderschat of genegeerd in proof-of-principle-demonstraties van nieuwe machine learning-toepassingen, ook omdat het meestal moeilijk te detecteren is.

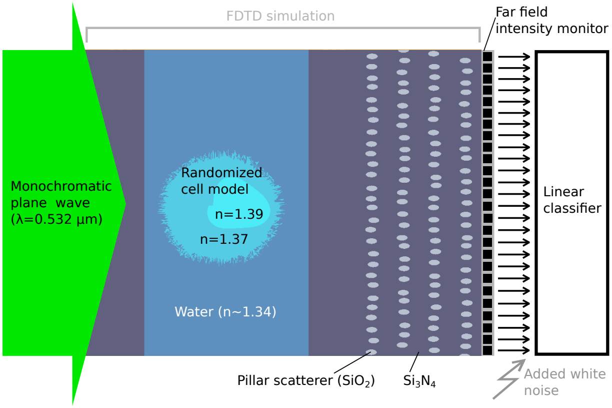
We hebben dit probleem grondig onderzocht en we hebben geprobeerd het op te lossen door verschillende benaderingen uit te voeren. Uiteindelijk realiseerden we ons echter dat we aanvullende gegevens van nieuwe metingen nodig hadden van onze partners, als referentie om te beoordelen in hoeverre de afwijkingen in meetomstandigheden van invloed waren op de training van onze classificator. Helaas was dit om technische redenen niet mogelijk.

2 Celclassificatie verbeterd door on-chip diëlektrische verstrooiers

Als tweede stap in dit doctoraatsonderzoek onderzochten we de voordelen van een interface tussen het microfluïdische kanaal en een fotonische microchip met daarin een verzameling microscopisch kleine diëlektrische pilaren die het licht dat in de chip is gekoppeld, verstrooien. Deze verstrooiers zijn bedoeld om de laserstraal, die door een stromende cel wordt gezonden, willekeurig te mengen voordat deze de beeldsensor bereikt. Aangezien de celinformatie hoofdzakelijk gecodeerd is in het faseprofiel van de laserbundel, vormt het corresponderende intensiteitspatroon dat door de beeldsensor wordt verkregen een niet-lineaire afbeelding van een dergelijk signaal. Daarom biedt de optische menging die wordt uitgevoerd door de beschouwde diëlektrische verstrooiers een manier om de complexiteit van deze niet-lineaire transformatie te verbeteren en te beheersen, wat op zijn beurt de rekenkracht van een eenvoudige lineaire classificator die rechtstreeks op de beeldpixelwaarden wordt toegepast, kan vergroten.

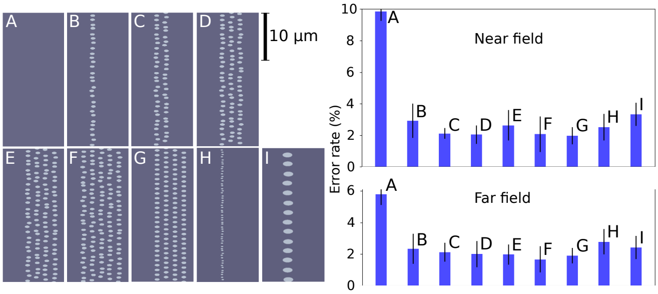
In feite komt deze classificatietechniek overeen met een op hardware gebaseerde extreme leermachinemethode, waarbij de combinatie van diëlektrische verstrooiers en beeldsensor de rol van het willekeurige niet-lineaire fysieke systeem op zich neemt dat de dimensionaliteit van het ingangssignaal kan vergroten. Het doel is om de berekening van niet-lineaire geëxtraheerde kenmerken (zoals de snelle Fourier-transformatie die in vorig onderzoek werd gebruikt) te vermijden door deze te vervangen door de praktisch onmiddellijke lichtvoortplanting door de verstrooiers. Aangezien de omzeilde niet-lineaire bewerkingen gewoonlijk veel hogere rekenkosten hebben dan de gewogen som die wordt uitgevoerd door de lineaire classificator, kan de algehele classificatiesnelheid aanzienlijk worden verhoogd.

We leverden een proof-of-concept demonstratie van deze techniek door middel van 2D eindige-verschil tijdsdomeinsimulaties (FDTD) (zie Fig. 3). Benadrukt moet worden dat het doel van deze simulaties niet is om de beschouwde flowcytometriemetingen nauwkeurig te beschrijven, maar om de soorten wiskundige bewerkingen die de celinformatie ondergaat tijdens het hologramverwervingsproces bij benadering te modelleren. We hebben de classificatie van 2D-celmodellen overwogen op basis van hun gemiddelde kerngrootte en, afzonderlijk, op basis van hun kernvorm. Om voldoende monsters te leveren om de machine learning classifier te trainen en te testen, hebben we duizenden simulaties uitgevoerd, met variabele celvorm, rotatie en positie.



Figuur 3: Schematische voorstelling van de classificatiepijplijn, inclusief een voorbeeld van 2D FDTD-simulatie. Een monochromatische vlakke golf treft een microfluïdisch kanaal dat een willekeurig celmodel in water bevat ($n_{H_2O} \approx 1.34$, $n_{\text{cytoplasm}} = 1.37$, $n_{\text{nucleus}} = 1.39$). Het voorwaarts verstrooide licht passeert een verzameling silica-verstrooiers ($n_{SiO_2} \approx 1.461$) ingebed in siliciumnitride ($n_{Si_3N_4} \approx 2.027$) en georganiseerd in lagen. De stralingsintensiteit wordt vervolgens opgevangen door een 1D verreveldmonitor, die is opgedeeld in pixels. Elke pixelwaarde wordt ingevoerd in een getrainde lineaire classificator (logistische regressie) die bestaat uit een gewogen som van de pixelwaarden.

We toonden aan dat het gebruik van diëlektrische verstrooiers de verkregen classificatienauwkeurigheid verdubbelde. Een nog grotere verbetering werd bereikt door te kijken naar UV-laserlicht of door het microfluïdische kanaal op te nemen in een Fabry-Pérot optische caviteit. Bovendien hebben we het gebruik van verschillende verschillende verstrooiingsconfiguraties onderzocht, zowel door de virtuele beeldsensor in het nabije veld als in de verreveldgebieden te plaatsen (Fig. 4). Interessant is dat de bereikte classificatieverbetering niet significant varieert met de beschouwde verschillen in de verstrooiingsconfiguratie.

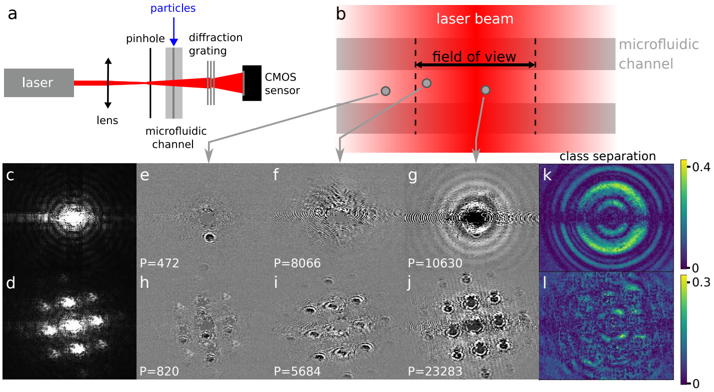


Figuur 4: Foutschattingen (blauwe balken, rechts) van celclassificatie op basis van de gemiddelde kernvorm, voor verschillende verstrooiingsconfiguraties (links) die in de simulaties worden gebruikt. De hoofdletters geven de overeenstemmende configuratie weer. De bovenste en onderste staafdiagrammen tonen respectievelijk de resultaten met betrekking tot de classificatie van het nabije veld en de patroonclassificatie in het verre veld. Voor elke verstrooierconfiguratie werden 7200 simulatieresultaten als monsters gebruikt.

3 Computationeel goedkope experimentele classificatie van microsferen

De laatste stap in dit doctoraatsonderzoek was om wat we geleerd hebben in de vorige beschouwingen toe te passen op een speciaal proof-of-concept flowcytometrie-experiment. In het bijzonder hebben we gekeken naar de classificatie van transparante microdeeltjes op basis van hun gemiddelde diameters (15.2 µm en 18.6 µm). Afbeeldingen van de interferentiepatronen van microdeeltjes werden verkregen met behulp van een zeer eenvoudige opstelling en een speciale meetbenadering werd gebruikt om het probleem van 'shortcut learning' als gevolg van afwijkingen in de meetomstandigheden te beheersen en te verhelpen. De ontwikkeling van de meetopstelling vergde relatief veel werk en testen, waaraan Emmanuel Gooskens en Jeremy Vatin deelnamen in het kader van respectievelijk hun afstudeeropdracht en stage.

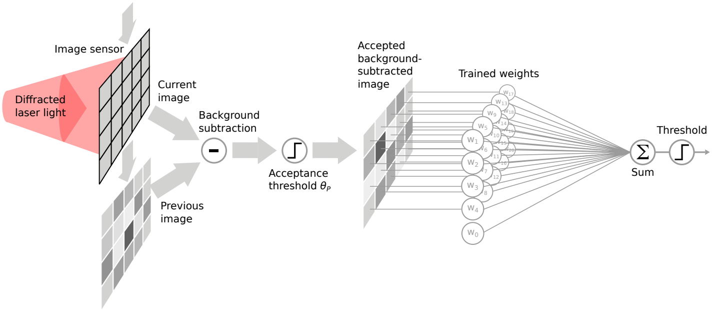
Bij de uiteindelijke realisatie van het experiment hebben we ook rekening gehouden met het plaatsen, tussen het microfluïdische kanaal en de beeldsensor, van een doorlatend diffractierooster voor optische menging, als vervanging voor de geïntegreerde diëlektrische verstrooiers die in de vorige FDTD-simulaties werden overwogen. De classificatietaak was echter wezenlijk anders t.o.v. degene die in onze FDTD-simulaties wordt overwogen. In het bijzonder, vanwege het grote gezichtsveld van onze cytometer, vertoonden de verworven interferentiepatronen een sterke variabiliteit als gevolg van deeltjesverplaatsing t.o.v. het verlichtingscentrum. Dit verhoogde de moeilijkheidsgraad van de classificatie aanzienlijk, maar het zorgde ook voor een hogere gevoeligheid voor cytometrie. Een schets van de opstelling en voorbeelden van verworven deeltjespatronen worden gegeven in Fig. 5.



Figuur 5: *a*: Schets van de toegepaste opstelling. Een microfluïdisch kanaal wordt verlicht door rode laserstraling gericht op een gaatje. De resulterende straal gaat door een doorlatend dubbel-assig diffractierooster (indien toegepast) en wordt opgevangen door een beeldsensor. *b*: Tekening van het verlichte microfluïdische kanaalgebied. Hoe groter de deeltjesafstand van het gezichtsveldcentrum, hoe zwakker het verkregen deeltjessignaal. *c*-*l*: met (onderste rij) en zonder (bovenste rij) het gebruik van een diffractierooster, voorbeelden van achtergrondpatroon (1^{st} kolom), achtergrond-afgetrokken deeltjespatronen met toenemende intensiteit (2^{nd} tot 4^{th} kolommen) en een kleurenkaart met de geschatte relevantie van elke pixel voor de classificatietaak (laatste kolom). Grijze pijlen suggereren een kwalitatief verband tussen de patroonvoorbeelden en de deeltjespositie t.o.v. het gezichtsveld getoond in *b*.

We gebruikten een eenvoudige en computationeel goedkope classificatiepijplijn, bestaande uit een combinatie van achtergrondaftrekking, achtergrondafbeelding weggooien en een lineaire classificatie toegepast op de beeldpixelwaarden (Fig. 6). Zonder een diffractierooster te gebruiken voor optische menging, kunnen we een bevredigend hoge classificatienauwkeurigheid (> 90%) bereiken voor verschillende resoluties van de opgenomen beelden, tot 32×26 pixels. Met deze specifieke beeldresolutie behaalde onze classificatiepijplijn de extreem lage uitvoeringstijd van 13μs op een gewone laptop. Dit is minstens een orde van grootte sneller dan de algoritmen die werden gerapporteerd in andere vergelijkbare werken over snelle deeltjes- of celclassificatie, die werden versneld met een GPU.

Interessant genoeg bereikten we vergelijkbare classificatienauwkeurigheden door het diffractierooster te gebruiken voor optische menging. Zelfs als dit niet



Figuur 6: Schets van de classificatiepijplijn voor machine learning. Storingspatronen worden geregistreerd door de beeldsensor. Het verschil tussen opeenvolgende afbeeldingen wordt berekend (aftrekken van de achtergrond), en de afbeeldingen die deeltjespatronen bevatten die niet intens genoeg zijn, worden weggegooid. Een lineaire classifier (trainbare gewogen som) wordt direct op de beeldpixels toegepast.

tot een classificatieverbetering leidde, hebben we bewezen dat onze classificatiemethode robuust is voor sterke vervorming van de verworven deeltjespatronen. Bovendien hebben we laten zien hoe shortcut learning kan worden gedetecteerd en behandeld in dit type flowcytometrie. Ten slotte hebben we flowcytometriebewerkingen gedemonstreerd met een aanzienlijk eenvoudigere opstelling, gemaakt van goedkopere componenten in vergelijking met conventionele implementaties.

4 Conclusie

In dit proefschrift onderzochten we de toepassing van de hardware-gebaseerde extreme learning machine-benadering om machine learning-classificatie van cellen of deeltjes in labelvrije beeldvormende flowcytometrie te versnellen. Door middel van FDTD-simulaties hebben we laten zien hoe on-chip diëlektrische scatterers kunnen worden gebruikt als hardwareversneller om de prestaties van een computationeel goedkope lineaire classificator die rechtstreeks op de beeldpixels wordt toegepast, aanzienlijk te verbeteren. Bovendien hebben we, rekening houdend met een vergelijkbare aanpak, een flowcytometrie-experiment ontwikkeld en getest om extreem snelle machine learning-classificatie van deeltjesinterferentiepatronen aan te tonen.

We hebben ook in detail onderzocht, en een praktische oplossing geboden voor, het ongrijpbare probleem van 'shortcut learning' veroorzaakt door langzame afwijkingen in meetomstandigheden, dat waarschijnlijk van invloed zal zijn op dit type machine learning-toepassing. Dat deden we door middel van machine learning-experimenten op hologrammen van witte bloedcellen die onze partners bij imec hadden verworven, en ook door ons eigen deeltjesclassificatie-experiment te beschouwen.

De in dit proefschrift ontwikkelde methodologie kan worden toegepast op bestaande high-throughput imaging flowcytometers om snelle online machine learning-operaties mogelijk te maken. Bovendien hebben we eenvoudige en computationeel efficiënte flowcytometrie gedemonstreerd met behulp van goedkope en compacte componenten.

English summary

In recent years, machine learning, and in particular deep learning, has been applied to address a wide variety of problems, allowing to automatically carry out tasks that are too difficult for traditional computing. However, popular and powerful models such as deep neural networks often present a relatively high computational cost, both in their training and in their application. By implementing part of a machine learning algorithm within dedicated hardware, large improvements in computational efficiency can be obtained. Nonetheless, it is particularly challenging to fabricate complex physical systems, e.g. a physical artificial neural network, that can be tuned, measured and modelled in enough detail and with enough precision to allow the application of conventional machine learning training for computationally efficient operations.

These challenges can be (at least partially) bypassed by combining random and nonlinear physical systems that only require limited tunability and observability, with simple and computationally cheap machine learning techniques based on linear operations, such as linear regressors or classifiers. Such an approach is usually referred to as the hardware-based version of reservoir computing (for dynamical hardware comprising feedback loops) or extreme learning machine methods (in the absence of feedback loops). These techniques have recently provided state-of-the-art performance in several types of applications and they require a much simpler and faster training compared to their conventional machine learning or deep learning counterparts. The role of the random nonlinear physical system is to map the input to a richer representation of higher dimensionality, in order to significantly boost the computational power of the subsequent linear machine learning model.

In this dissertation we apply this approach, exploiting the extreme processing speed of photonic systems to classify microscopic objects, such as biological cells or microparticles, illuminated while flowing in a microfluidic channel. I.e., we aim to simplify and speed up machine learning operations in label-free microflow cytometry. Indeed, the computational cost of traditional algorithms often acts as bottleneck in this type of application, significantly limiting the throughput of online operations, such as cell sorting.

5 Efficient white blood cell classification in imaging label-free microflow cytometry

We started this PhD research with a collaboration with researchers from IMEC (R&D hub for nano- and digital technologies) who developed a label-free imaging flow cytometer for high-throughput sorting of white blood cells, based on a lens-free inline digital holographic microscopy method. In simpler words, they made three different types of white blood cells (monocytes, T cells and granulocytes) flow in a transparent microfluidic channel, and they recorded the interference patterns projected by these cells when illuminated with visible laser light shined through a microscopic aperture (see Fig. 7). In digital holographic microscopy, the acquired patterns (called holograms) are fed into a computationally expensive image reconstruction algorithm to obtain the cell image. However, our goal consisted of directly applying a machine learning algorithm to the raw hologram, greatly reducing the required computation for cell classification.

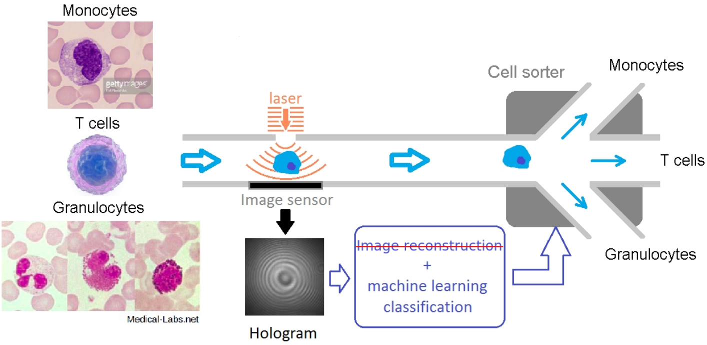


Figure 7: The main functionalities of the flow cytometer (developed by imec) that generated the considered holograms. Three different white blood cell types are made to flow in a microfluidic channel. Visible laser light is shined through an aperture perpendicularly to the channel and the resulting diffraction pattern is projected on an image sensor. The presence of a flowing cell, whose light absorption is negligible, modifies the interference pattern acquired by the sensor. The obtained image, called cell hologram, carries information about the 3D refractive index structure of the cell, which can be used to reconstruct the cell image (digital holographic microscopy). This operation is computationally expensive, thus we directly applied a machine learning classifier to raw cell holograms.

In order to further decrease the computational cost of cell classification, and thus to increase the maximum throughput of cell sorting, we had to dramatically decrease the large dimensionality (given by around a million of pixel values) of the cell hologram images, by extracting fewer relevant and non-redundant features. We did so by transforming the original 2D patterns into 1D patterns through pixel summation along the channel direction, approximating the result obtained using a 1D image sensor. Afterwards, we applied suitable translation-invariant functions (auto-correlation and fast Fourier transform) and we fed the resulting values to a linear classifier based on logistic regression (Fig. 8). Such a feature extraction pipeline was meant to reduce the noise introduced by the variability of the cell pattern position w.r.t. the image center.

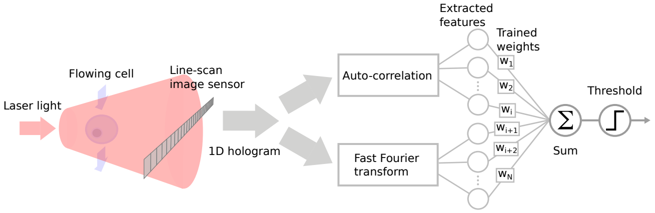


Figure 8: The proposed classification pipeline. We obtain 1D holograms by summing 2D hologram pixels along the flow direction, which is an approximation of the use of a line-scan image sensor. Suitable auto-correlation and fast Fourier transform values are calculated from the 1D hologram (feature extraction). These features are then weighted and summed by a linear classifier (logistic regression). An optimal set of weights for each class is learned during training. The weighted sum with highest outcome indicates the cell class recognized by the classifier.

This computationally efficient classification pipeline showed satisfying classification performance when applied to the data provided by our partners. However, we discovered that the obtained results were heavily biased by the correlation between slow drifts in measurement conditions (affecting the laser beam illuminating the cells) and the cell classes. In particular, holograms of each cell type had been acquired in a single long measurement, one after another, and therefore the generated data had been marked by the drifting measurement conditions. This allowed our machine learning algorithm to easily learn the image classification exploiting the measurement condition information, instead of the actual cell information. Such a machine learning issue, called shortcut learning or dataset bias, is often underestimated or ignored in proof-of-principle demonstrations of new machine learning applications, also because it is usually difficult to detect.

We investigated this problem thoroughly and we attempted to overcome it trying various approaches. However, we finally realized that we needed additional data from our partners from new measurements, as a reference to asses how much the drifts in measurement conditions affected the training of our classifier. Unfortunately, this was not possible because of technical reasons.

6 Cell classification improved by on-chip dielectric scatterers

As a second step in this PhD research we investigated the advantages of interfacing the microfluidic channel with a photonic microchip containing a collection of microscopic dielectric pillars that scatter the light coupled in the chip. These scatterers are intended to randomly mix the laser beam transmitted through a flowing cell before it reaches the image sensor. Since the cell information is mainly encoded in the phase profile of the laser beam, the corresponding intensity pattern acquired by the image sensor constitutes a nonlinear mapping of such a signal. Therefore, the optical mixing performed by the considered dielectric scatterers provides a way to enhance and control the complexity of this nonlinear transformation, which in turn may boost the computational power of a simple linear classifier directly applied to the image pixel values.

In fact, this classification technique corresponds to a hardware-based extreme learning machine method, where the combination of dielectric scatterers and image sensor takes the role of the random nonlinear physical system which can expand the dimensionality of the input signal. The aim is to avoid the computation of nonlinear extracted features (such as the fast Fourier transform employed in the previous investigation) by replacing it with the practically instantaneous light propagation through the scatterers. Since the bypassed nonlinear operations have usually a much higher computational cost than the weighted sum performed by the linear classifier, the overall classification speed can be significantly enhanced.

We provided a proof-of-concept demonstration of this technique by means of 2D finite-difference time-domain simulations (see Fig. 9). It should be stressed that the aim of these simulations is not to accurately describe the considered flow cytometry measurements, but just to approximately model the types of mathematical operations that the cell information undergo during the hologram acquisition process. We considered the classification of 2D cell models on the basis of their average nucleus size and, separately, on the basis of their nucleus shape. In order to provide enough samples to train and test the machine learning classifier, we performed thousands of simulations, with variable cell shape, rotation and position.

We showed that the use of dielectric scatterers doubled the obtained classification accuracy. An even larger improvement was achieved by considering UV laser light or by including the microfluidic channel in a Fabry-Pérot optical cavity. Moreover, we explored the employment of several different scatterer configurations both placing the virtual image sensor in the near field and in the far field regions (Fig. 10). Interestingly, the achieved classification improvement does not significantly vary with the considered differences in scatterer configuration.

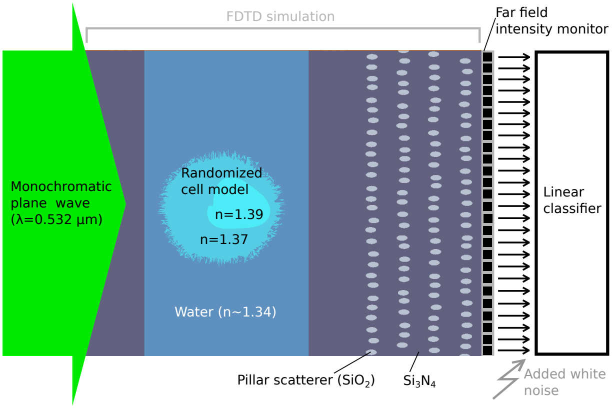


Figure 9: Sketch of the classification pipeline, including an example of a 2D FDTD simulation. A monochromatic plane wave impinges on a microfluidic channel containing a randomized cell model in water ($n_{H_2O} \approx 1.34$, $n_{\text{cytoplasm}} = 1.37$, $n_{\text{nucleus}} = 1.39$). The forward scattered light passes through a collection of silica scatterers ($n_{SiO_2} \approx 1.461$) embedded in silicon nitride ($n_{Si_3N_4} \approx 2.027$) and organized in layers. The radiation intensity is then collected by a 1D far-field monitor, which is divided into bins (pixels). Each pixel value is fed into a trained linear classifier (logistic regression) that consists of a weighted sum of the pixel values.

7 Computationally cheap experimental classification of microspheres

The final step in this PhD research was to apply what we learned in the previous investigations to a dedicated proof-of-concept flow cytometry experiment. In particular, we considered the classification of transparent microparticles on the basis of their average diameters (15.2 µm and 18.6 µm). Images of the microparticle interference patterns were acquired using a very simple setup and a dedicated measurement approach was employed to control and overcome the shortcut learning issue due to drifts in measurement conditions. The setup and measurement development required a relatively large amount of work and tests, in which Emmanuel Gooskens and Jeremy Vatin took part, respectively in the context of their master thesis project and internship.

In the final realization of the experiment, we also considered the interposition, between the microfluidic channel and the image sensor, of a transmissive diffraction grating for optical mixing, as a substitute for the integrated dielectric scatterers considered in the previous FDTD simulations. The classification task was however essentially different w.r.t. the one considered in our FDTD

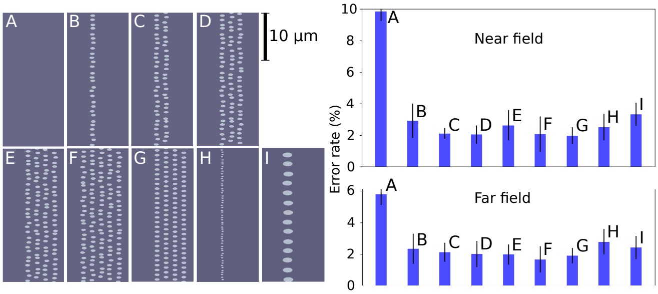


Figure 10: Error estimates (blue bars, on the right) of cell classification on the basis of the average nucleus shape, for different scatterer configurations (on the left) employed in the simulations. The capital letters show the configuration-error correspondence. The upper and the lower bar plots respectively show the results concerning the near-field and the far-field pattern classification. For each scatterer configuration, 7200 simulation results were employed as samples.

simulations. In particular, because of the large field-of-view of our cytometer, the acquired interference patterns presented a strong variability due to particle displacement w.r.t. the illumination center. This significantly increased the classification difficulty, but it also allowed for a higher cytometry sensitivity. A sketch of the setup and examples of acquired particle patterns are provided in Fig. 11.

We employed a simple and computationally cheap classification pipeline, consisting of a combination of background subtraction, background image discard and a linear classifier applied to the image pixel values (Fig. 12). Without using a diffraction grating for optical mixing, we could achieve satisfyingly high classification accuracy (> 90%) for different resolutions of the recorded images, down to 32×26 pixels. Considering this particular image resolution, our classification pipeline achieved the extremely low execution time of 13µs on a common laptop. This is at least one order of magnitude faster than the algorithms reported in other comparable works about fast particle or cell classification, which were accelerated by a GPU.

Interestingly, we achieved similar classification accuracies by using the diffraction grating for optical mixing. Even if this did not lead to a classification improvement, we showed that our classification method is robust to heavy distortion of the acquired particle patterns. Moreover, we demonstrated how shortcut learning can be detected and treated in this type of flow cytometry. Finally, we showed flow cytometry operations with a substantially simpler setup, made of cheaper components compared to conventional implementations.

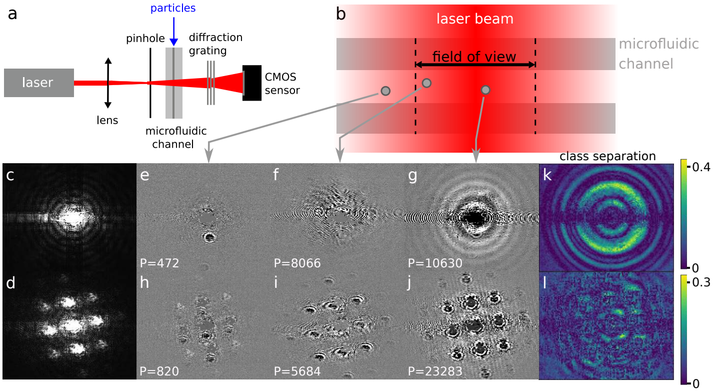


Figure 11: *a*: Sketch of the employed setup. A microfluidic channel is illuminated by red laser radiation focused on a pinhole. The resulting beam passes through a transmissive double-axis diffraction grating (when employed) and is captured by an image sensor. *b*: drawing of the illuminated microfluidic channel region. The larger the particle distance from the field-of-view center, the weaker the acquired particle signal. *c-l*: with (bottom row) and without (top row) the use of a diffraction grating, examples of background pattern (1^{st} column), background-subtracted particle patterns with increasing intensity (2^{nd} to 4^{th} columns) and a colormap showing the estimated relevance of each pixel for the classification task (last column). Grey arrows suggest a qualitative link between the pattern examples and the particle position w.r.t. the field-of-view shown in *b*.

8 Conclusion

In this dissertation we investigated the application of a hardware-based extreme learning machine approach to speed up machine learning classification of cells or particles in label-free imaging flow cytometry. By means of FDTD simulations, we showed how on-chip dielectric scatterers can be used as a hardware accelerator to significantly improve the performance of a computationally cheap linear classifier directly applied to the image pixels. Moreover, considering a similar approach, we developed and tested a flow cytometry experiment to demonstrate extremely fast machine learning classification of particle interference patterns.

We also investigated in detail, and provided a practical solution for, the elusive problem of shortcut learning caused by slow drifts in measurement conditions, which is likely to affect this type of machine learning application. We did so by means of machine learning experiments on white blood cell holograms acquired by our partners at imec, and also by considering our own particle classification experiment.

The methodology developed in this dissertation can be applied to existent

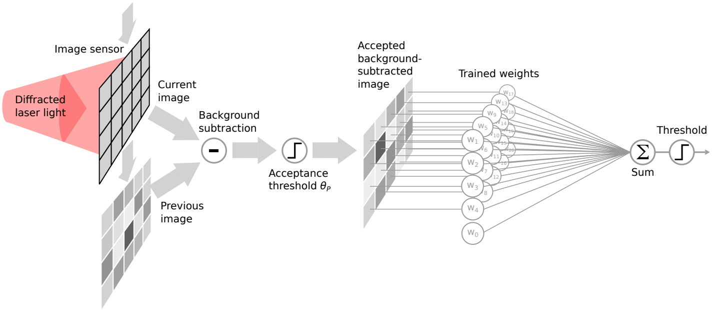


Figure 12: Sketch of the machine-learning classification pipeline. Interference patterns are acquired by the image sensor. The difference between consecutive images is calculated (background subtraction), and those images that do not contain intense enough particle patterns are discarded. A linear classifier (trainable weighted sum) is directly applied to the image pixels.

high-throughput imaging flow cytometers to allow for fast online machine learning operations. Moreover, we demonstrated simple and computationally efficient flow cytometry using cheap and compact components.

0
Introduction

Accelerating machine learning with random physical systems

In recent years, machine learning, and in particular deep learning, has been applied to address a wide variety of problems, allowing to automatically carry out tasks that are too difficult for traditional computing. However, popular and powerful models such as deep neural networks often present a relatively high computational cost, both in their training and in their application. By implementing part of a machine learning algorithm within dedicated hardware, large improvements in computational efficiency can be obtained [1]. Nonetheless, it is particularly challenging to fabricate complex physical systems, e.g. a physical artificial neural network, that allow the application of conventional machine learning training for computationally efficient operations. In particular, it is difficult to build systems that can be tuned, measured and modelled in enough detail and with enough precision.

These challenges can be (at least partially) bypassed by combining random and nonlinear physical systems that only require limited tunability and observability, with simple and computationally cheap machine learning techniques based on linear operations, such as linear regressors or classifiers. Such an approach is usually referred to as the hardware-based version of *reservoir computing* [2] (for dynamical hardware comprising feedback loops) or *extreme learning machine* methods [3, 4] (in the absence of feedback loops). The software versions of these techniques have recently provided state-of-the-art performance in several types of applications and they require a much simpler and faster training compared to their conventional machine learning or deep learning counterparts. In hardware-based implementations, the role of the random nonlinear physical system is to map the input to a richer representation of higher dimensionality, in order to significantly boost the computational power of the subsequent linear machine learning model. We will explain the basic principles at the root of this approach in Chapter 1.

In this dissertation we apply such an approach, exploiting the extreme processing speed and high parallelism of diffractive optical layers (such as diffraction gratings), to simplify and speed up machine learning classification of cells and particles in flow cytometry.

A relevant example of how practically instantaneous random projections performed by scattering optical media (such as diffusive media) can be used to accelerate machine learning applications is provided by [5]. The authors demonstrate the classification of handwritten digits images using the proposed optical hardware accelerator, which corresponds to a hardware implementation of the popular kernel machine approach. In particular, they spatially encode the sample images in the amplitude of a laser beam by means of a digital micromirror device. The output radiation is then focused on a random scattering medium and the transmitted light is recorded by an image sensor. A linear classifier is then trained on the acquired interference patterns.

In this dissertation we consider a similar approach, where the input information is instead spatially encoded in the phase of a laser beam by its transmission through a cell or a transparent microparticle. Moreover, we implement the optical mixing operations by means of more ordered optical scattering media, such as diffraction gratings.

Remarkably, the combination of a reconfigurable optical diffractive layer with an image sensor recently enabled powerful hardware accelerators for deep neural networks, outperforming state-of-the-art GPUs in terms of computational throughput and energy efficiency [6].

1 Label-free imaging microflow cytometry

Flow cytometers are instruments that are able to analyze and characterize large numbers of suspended biological cells and microparticles one by one, while these are flowing at high speed through a measuring device [7]. In traditional flow cytometers, the moving particles are illuminated, usually by a laser, and the corresponding forward and/or side-scattering intensities are measured, together with the fluorescent emission of selectively attached probes (Fig. 1). These devices are widely used to investigate the structure and the chemical composition of large populations of cells in many applications concerning life science and clinical diagnosis. Moreover, they also find diverse applications in industrial and environmental engineering fields, e.g. in measuring bacteria viability [8] or water quality [9].

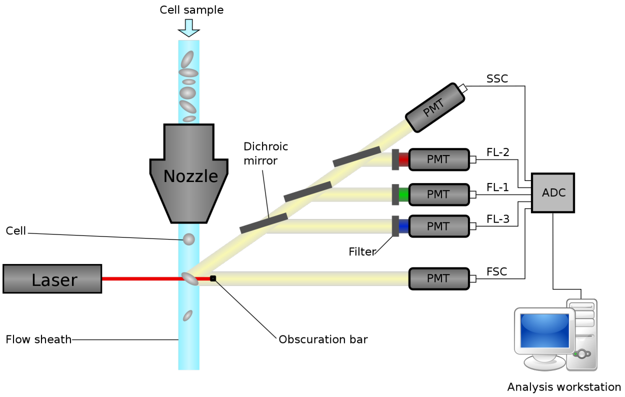


Figure 1: Schematic of a conventional flow cytometer. Cell classification is performed considering various detected features of the light scattered by a flowing cell: forward scattering (FSC), side scattering (SSC) and different fluorescent wavelengths generated by the fluorescent labels that are selectively attached to the cells. Figure adapted from [10], subject to Creative Commons Attribution 3.0 Unported license [11].

Although flow cytometers were constantly innovated upon in the last few decades, their use is still limited by high cost, complexity and size [12]. Let us now follow a path through some of the recent approaches proposed by the scientific and engineering community to overcome these limitations, in order to contextualize the presented work.

To begin with, the integration of *microfluidic* systems on a chip allows for a great reduction in cytometers' cost and size, which is particularly appealing for point-of-care applications [12]. Furthermore, the integration with other lab-on-chip devices provides the opportunity for increased automation and for scalable parallelization of particle analysis, potentially multiplying the overall device throughput [13–15].

While the use of fluorescent labels in flow cytometry provides a powerful instrument to discriminate between different cell populations at high throughput (even exceeding $100,000$ cells/s [14]), the application of fluorescent stains (also called *labels*) often hinders live cell analysis, for instance because of cytotoxicity, and requires dedicated effort and cost [16]. Two increasingly common approaches to enable accurate and relatively fast label-free analysis while improving detection sensitivity are given by *electrical impedance detection* and *imaging flow cytometry* [12]. This work mostly focuses on the latter, whose main advantage is the acquisition of detailed spatial information that can be used both for morphology-based detection and for human visualization as in traditional microscopy. On the other hand, the operational speed of camera-based cytometers is limited by the acquisition frame rate and motion blur, providing single-channel throughputs up to around 1000 cells/s when single cells are captured [17].

This limitation can be overcome, at the cost of increasing system and instrumentation complexity, by encoding optical spatial information into a temporal sequence that is measured by a single photodetector. Such a technique, referred to as *optofluidic time-stretch microscopy*, combines the wide spectral bandwidth of a femtosecond pulse laser with both temporal and spatial dispersive optical elements, achieving label-free single cell imaging at a very high throughput, up to around $100,000$ cells/s [18–20]. It should be stressed that achieving such high throughputs allows for statistically relevant screening of large cell populations, which is key in many biological studies and clinical diagnoses.

Let us see in further detail the basic functioning of optofluidic time-stretch microscopy, with reference to Fig. 2. A broadband pulse laser is employed as light source, and its output is inserted into a temporal disperser (e.g. a dispersive fiber), through which the different frequency components of a pulse travel at different velocities. I.e., the optical pulses are time-stretched as their frequency components exit the time disperser at different times. Afterwards, the time-stretched pulses impinge on a first spatial disperser (e.g. a diffraction grating), which maps the different frequency components of a pulse along a line perpendicular to the employed microfluidic channel. The temporally and spatially stretched pulses are then focused by an objective lens onto the microfluidic channel, inside which the cells flow at high speed. Intuitively, each pulse can perform an extremely fast line scan of a passing cell, encoding the corresponding spatial information into the different frequency components. Therefore, several consecutive pulses can perform a complete 2D scan of a flowing cell. The pulses are then projected via another objective lens on a second spatial disperser, of the same type of the first one, so that the different frequency components are spatially realigned and captured by a single photodetector. Finally, 2D cell images can be obtained by digitally stacking the temporal segments of the acquired signal that correspond to different laser pulses.

Because of its ability to perform single-cell label-free analysis at an extremely high throughput, optofluidic time-stretch microscopy has been employed in several scientific and industrial applications, such as cancer cell detection, drug

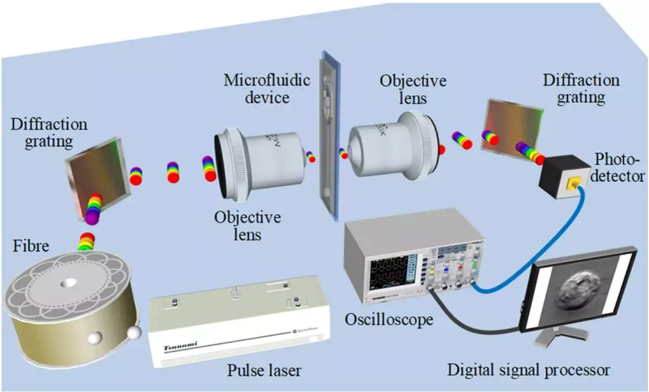


Figure 2: Sketch of an optofluidic time-stretch microscope for high-throughput label-free imaging flow cytometry. Figure adapted from [21], subject to Attribution 4.0 International license [22].

screening and phytoplankton classification.

Another promising option to overcome the throughput limitation imposed by the camera frame rate and motion blur is to perform the cell analysis in parallel employing multiple particle streams. The scalability of such an approach can be greatly enhanced by "lab-on-chip" technologies [13, 15]. In particular, the idea of imaging several microfluidic channels (working in parallel) at the same time with a single image sensor is especially interesting. In [13], this is done by stacking a 4×4 array of pinholes, microfluidic channels and diffractive microlenses, achieving a throughput of around $20,000$ images per second.

Most of the methods investigated and developed in this dissertation can be straightforwardly applied to both the parallel channel approach and to the optofluidic time-stretch microscopy for high-throughput online analysis.

An interesting example of how random photonic operations can significantly improve high-speed implementations such as optofluidic time-stretch microscopy for image classification, is presented in [23]. The authors propose to add further operations in the optical domain in order to avoid the employment of complex and costly high-speed electronic components at the end of the measurement pipeline. In particular, before the optical signal is detected, it is inserted into a photonic integrated recurrent neural network with fixed random connections for hardware-based reservoir computing. The obtained optical output signals, which map the input to a higher dimensional representation, are measured by suitable photodiodes and the resulting electronic signals are fed into a linear classifier, that can thus be trained to carry out the pattern detection task.

2 Machine learning for automatic online cell detection

The automatic analysis of digital images is a powerful and versatile tool, but it is usually computationally expensive and memory hungry due to the high data dimensionality given by the large number of pixels. In high-throughput imaging cytometers, the huge number of stored images and the required processing time are an issue [17], even more when compact and cheap applications are targeted, e.g. point-of-care. Furthermore, online image analysis often requires a too high computational power such that real-time cell sorting cannot easily be done. Several machine learning approaches have recently been proposed to automatically analyze the big amounts of data generated by label-free imaging flow cytometry [16, 24–30], although in most of them the image processing is carried out offline. Exceptions are [26, 27, 31], where single-particle classifications respectively took < 1 ms, 0.2 ms and 3.6 ms when accelerated by a GPU. These were applied on images of respectively 21x21 and 32x32 pixels in the first 2 works, while in the third the original time-stretch-microscope resolution (which was not explicitly mentioned) was reduced by a factor of 40. However, these execution times are still far from enabling real-time classification for state-of-the-art high throughputs of around $100,000$ cells/s, especially if higher resolutions are required to distinguish specific cell features. Throughout this dissertation we will address this specific problem, by investigating and developing dedicated machine learning approaches for computationally efficient cell or microparticle classification.

A crucial operation in machine learning applications is to obtain an accurate and unbiased *ground truth*, which in classification problems is the true class of the samples used to train and test the employed models. In the demonstrations of cell classification within label-free imaging flow cytometry, the safest way to obtain the ground truth is by making cells of different classes flow together and to employ a trusted alternative classification method to infer the correct class of the imaged cells.

An example that is very relevant to this work is given by [30], where several types of white blood cells flowing together were classified at the same time using a conventional and trusted technique (manual gating, using fluorescent labels) and the new methods under evaluation (4 types of machine learning classifiers applied to label-free images). In the classification of 8 different types of white blood cells, the highest accuracy achieved was 77.8% and 70.3%, respectively for classical machine learning (based on feature engineering) and for deep learning. These results show that the employed algorithms are effective, but also that it is difficult to achieve very high accuracies in this type of classification task. Moreover, this exemplifies how deep learning might not always the best option for tackling these problems.

However, implementing an alternative and trusted system that provides the ground truth in the same cytometer (when possible) is often neither convenient nor practical. Indeed, in many other demonstrations, the ground truth is obtained separately, before the flow cytometry operations under investigation, and cells belonging to different classes are measured at different times. In this dissertation we will show how this practice can degrade the machine learning training and inflate the classification performance evaluation, due to the influence of slowly drifting measurement conditions on the recorded sample images. This constitutes a specific case of an elusive and often underestimated machine learning issue, referred to as *shortcut learning* or dataset bias [32]. After a detailed investigation of this problem in the context of the considered applications, we will develop a methodology to keep shortcut learning under control.

Moreover, we will consider the *digital in-line holographic microscopy* method as the source of image samples, which constitutes a further step towards significantly cheaper and more compact imaging flow cytometers [28, 33, 34]. This technique enables lens-free microscopy, by acquiring unfocused images that are subsequently reconstructed in software. In particular, a laser beam is directed onto a small pinhole, whose diameter is comparable with the laser wavelength, so that it approximately acts as a point source emitting a spherical wave (Fig. 3). The resulting radiation illuminates an object positioned close (at few millimeters distance) to the pinhole projecting a geometrically magnified diffraction pattern (called also *hologram*) on an image sensor.

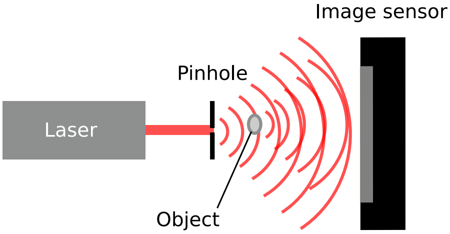


Figure 3: Working principle of the hardware components in digital in-line holographic microscopy. The geometrical magnification of the diffraction pattern projected by the object is due to the emanation of approximately spherical waves from the pinhole. From the acquired hologram a focused image can be obtained in software through image reconstruction.

The execution time of image reconstruction usually spans from a few tenths of a second to several seconds depending on the algorithm and on the image resolution [27, 35]. However, in principle machine learning algorithms do not require a focused image, as they can learn the desired operations directly from the acquired holograms, as long as they contain relevant cell information. The idea of bypassing the computationally expensive image reconstruction and performing machine learning operations directly on raw interference patterns was proposed in the past [36] and recently experimentally applied [27], in the context of white blood cell classification in label-free imaging flow cytometry. The work described in this dissertation is also based on this approach.

3 Objectives

In this dissertation, our general aim is to exploit the advantages brought by machine learning to simplify automatic classification in label-free imaging flow cytometry, in terms of computation, usage and required components. In particular, we focus on the use of linear classifiers (based on logistic regression), which are much simpler to train and computationally cheaper in their application than other more complex machine learning models, such as artificial neural networks. We mainly consider their direct application on the interference patterns (or on their background subtracted versions) projected by cells or microparticles when illuminated by a visible laser beam. This approach guarantees a computational advantage and higher versatility w.r.t. more conventional methods, where intermediate nonlinear functions of the acquire images are computed, usually chosen on the basis of specific domain knowledge. Instead of doing so, we aim to act on the optical hardware components, in the attempt to optimize the recorded interference patterns specifically for classification with linear machine learning models. This includes the interposition of optical diffractive layers between the illuminated object and the camera, which enables practically instantaneous optical mixing operations. The versatility of this approach also allows to employ simpler and cheaper components for flow cytometry, and reduces the need for domain knowledge and machine learning expertise in its application.

Another objective of this dissertation is to investigate and find practical solutions to the problem of shortcut learning due to slow drifts in measurement conditions. Indeed, the high sensitivity of the data generated by imaging flow cytometry to slight changes in the optical path, can easily and elusively bias the training and testing of a machine learning classifier, producing inflated performance estimations and undermining its real applicability.

4 Dissertation outline

In Chapter 1 we cover the basic machine learning concepts and techniques employed throughout this dissertation. We do so by considering mock-up classification tasks in order to provide practical and intuitive explanations.

In Chapter 2 we focus on the development of a software machine learning approach to classify holograms projected by three types of white blood cells at a low computational cost. The holograms were provided by our collaborators from imec (R&D hub for nano- and digital technologies) and were generated by their prototype of a label-free imaging flow cytometer based on in-line digital holographic microscopy. In our attempt to accomplish this task, we encounter and investigate the problem of shortcut learning due to slow drifts in measurement conditions.

In Chapter 3 we consider the interposition of on-chip microscopic optical scatterers between the microfluidic channel and the image sensor. In particular, we investigate if the corresponding optical mixing can improve the performance of a linear classifier applied to the acquired interference patterns. To generate the required samples we perform thousands of 2D FDTD optical simulations, which approximately model the considered optical processes.

In Chapter 4 we go through the development of a proof-of-concept experiment, where we build and employ a simple imaging microflow cytometer to demonstrate computationally efficient microparticle classification. In particular, we address the shortcut learning problem encountered in Chapter 2, and we investigate the application of optical extreme learning machine approaches similar to the one considered in Chapter 3.

Finally, in Chapter 5 we discuss the results obtained in the final version of the experiment developed in Chapter 4. We especially focus on the achievement of particle classification at a low computational cost and on overcoming the shortcut learning issue.

5 Publications

Publications in international journals

[1] Alessio Lugnan, Emmanuel Gooskens, Jeremy Vatin, Joni Dambre, and Peter Bienstman. *Machine learning issues and opportunities in ultrafast particle classification for label-free microflow cytometry.* Scientific Reports, 10(1):1–13, 2020.

[2] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated pillar scatterers for speeding up classification of cell holograms.* Optics Express, 25(24):30526–30538, 2017.

[3] Alessio Lugnan, Andrew Katumba, Floris Laporte, Matthias Freiberger, Stijn Sackesyn, Chonghuai Ma, Emmanuel Gooskens, Joni Dambre, and Peter Bienstman. *Photonic neuromorphic information processing and reservoir computing.* APL Photonics (invited), 5:020901, 2020.

[4] Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, Joni Dambre, and Peter Bienstman. *Neuromorphic computing based on silicon photonics and reservoir computing.* IEEE Journal on Selected Topics in Quantum Electronics (invited), 24(6):8300310, 2018.

Publications in international conferences

[5] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated dielectric scatterers for fast optical classification of biological cells.* SPIE Photonics Europe, 10689(07):1–7, 2018.

[6] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated dielectric scatterers for speeding up classification of cell diffraction patterns.* 20th International Conference on Transparent Optical Networks (invited), 2018.

[7] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Numerical investigation of integrated dielectric pillars to simplify machine learning classification of cells.* 23rd Annual Symposium of the IEEE Photonics Benelux Chapter, 2018.

[8] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated pillar scatterers for speeding up classification of cell holograms through a RC-like machine learning approach.* Workshop on Dynamical Systems and Brain-inspired Information Processing, 2017.

[9] Floris Laporte, Andrew Katumba, Matthias Freiberger, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, Emmanuel Gooskens, Joni Dambre, and Peter Bienstman. *Photonic Reservoir Computing (invited).* Photonic Integration Week, 2020.

[10] Peter Bienstman, Joni Dambre, Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, and Emmanuel Gooskens. *Non-linear signal equalisation using silicon photonic reservoir computing (invited).* ECOC machine learning workshop, 2019.

[11] Peter Bienstman, Joni Dambre, Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, and Chonghuai Ma. *Neuromorphic information processing using silicon photonics (invited).* SPIE Optics and Photonics, 2019.

[12] Peter Bienstman, Joni Dambre, Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, and Emmanuel Gooskens. *Silicon photonics reservoir computing at 32 Gbit/s (invited).* 5th Workshop on Dynamical Systems and Brain-Inspired Information Processing, 2019.

[13] Peter Bienstman, Joni Dambre, Andrew Katumba, Matthias Freiberger, Floris Laporte, and Alessio Lugnan. *Photonic reservoir computing: a brain-inspired approach for information processing (invited).* In Optical Fiber Communication Conference, pages M4F–4. Optical Society of America, 2018.

[14] Peter Bienstman, Joni Dambre, Andrew Katumba, Matthias Freiberger, Floris Laporte, and Alessio Lugnan. *Silicon photonics for neuromorphic information processing.* In Optical Data Science: Trends Shaping the Future of Photonics, volume 10551, page 105510K. International Society for Optics and Photonics, 2018.

[15] Floris Laporte, Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Novel photonic reservoir computing architectures.* In Workshop on Dynamical systems and Brain-Inspired Information Processing, page 1, 2017.

[16] Andrew Katumba, Floris Laporte, Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated-photonics implementation of reservoir computing neural networks (invited).* In Workshop Machine Learning@ ECOC 2017, pages 1–1, 2017.

Patents

[17] Peter Bienstman, Alessio Lugnan, and Floris Laporte. *Object classification system and method.* Patent Application No. PCT/EP2018/063854, Filed May 26, 2018

[18] Peter Bienstman, Floris Laporte, and Alessio Lugnan. *Mixing wave based computing.* Patent Application No. PCT/EP2018/063855, Filed May 26, 2018

Book Chapters

[19] Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, Joni Dambre, and Peter Bienstman *Integrated on-chip reservoirs.* Photonic Reservoir Computing: Optical Recurrent Neural Networks, De Gruyter, 2019.

References

[1] Vivienne Sze, Yu-Hsin Chen, Joel Emer, Amr Suleiman, and Zhengdong Zhang. *Hardware for machine learning: Challenges and opportunities.* In 2017 IEEE Custom Integrated Circuits Conference (CICC), pages 1–8. IEEE, 2017.

[2] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. *Recent advances in physical reservoir computing: A review.* Neural Networks, 115:100–123, 2019.

[3] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. *Trends in extreme learning machines: A review.* Neural Networks, 61:32–48, 2015.

[4] Weipeng Cao, Xizhao Wang, Zhong Ming, and Jinzhu Gao. *A review on neural networks with random weights.* Neurocomputing, 275:278–287, 2018.

[5] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala. *Random projections through multiple optical scattering: Approximating kernels at the speed of light.* In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6215–6219. IEEE, 2016.

[6] Tiankuang Zhou, Xing Lin, Jiamin Wu, Yitong Chen, Hao Xie, Yipeng Li, Jingtao Fan, Huaqiang Wu, Lu Fang, and Qionghai Dai. *Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit.* Nature Photonics, pages 1–7, 2021.

[7] Aysun Adan, Günel Alizada, Yağmur Kiraz, Yusuf Baran, and Ayten Nalbant. *Flow cytometry: basic principles and applications.* Crit. Rev. Biotechnol., 37(2):163–176, mar 2017.

[8] Martin G. Wilkinson. *Flow cytometry as a potential method of measuring bacterial viability in probiotic products: A review.* Trends Food Sci. Technol., 78:1–10, aug 2018.

[9] Hannah R. Safford and Heather N. Bischel. *Flow cytometry applications in water treatment, distribution, and reuse: A review.* Water Res., pages 110–133, mar 2019.

[10] *Wikimedia Commons: Schematic of a flow cytometer, from sheath focusing to data acquisition.* https://commons.wikimedia.org/wiki/File:Cytometer.svg. Accessed: 2021-04-29.

[11] *Creative Commons Attribution: Attribution 3.0 Unported.* https://creativecommons.org/licenses/by/3.0/deed.en. Accessed: 2021-04-29.

[12] Ruey-Jen Yang, Lung-Ming Fu, and Hui-Hsiung Hou. *Review and perspectives on microfluidic flow cytometers.* Sensors and Actuators B: Chemical, 266:26–45, 2018.

[13] Ethan Schonbrun, Sai Siva Gorthi, and Diane Schaak. *Microfabricated multiple field of view imaging flow cytometry.* Lab on a Chip, 12(2):268–273, 2012.

[14] Y. J. Fan, Y. C. Wu, Y. Chen, Y. C. Kung, T. H. Wu, K. W. Huang, H. J. Sheen, and P. Y. Chiou. *Three dimensional microfluidics with embedded microball lenses for parallel and high throughput multicolor fluorescence detection.* Biomicrofluidics, 7(4):1–13, 2013.

[15] Liesbet Lagae, Dries Vercruysse, Alexandra Dusa, Chengxun Liu, Koen de Wijs, Richard Stahl, Geert Vanmeerbeeck, Bivragh Majeed, Yi Li, and Peter Peumans. *High throughput cell sorter based on lensfree imaging of cells.* In 2015 IEEE International Electron Devices Meeting (IEDM), pages 13–3. IEEE, 2015.

[16] Thomas Blasi, Holger Hennig, Huw D. Summers, Fabian J. Theis, Joana Cerveira, James O. Patterson, Derek Davies, Andrew Filby, Anne E. Carpenter, and Paul Rees. *Label-free cell cycle analysis for high-throughput imaging flow cytometry.* Nat. Commun., 7, jan 2016.

[17] Yuanyuan Han, Yi Gu, Alex Ce Zhang, and Yu Hwa Lo. *Review: Imaging technologies for flow cytometry.* Lab Chip, 16(24):4639–4647, 2016.

[18] K. Goda, K. K. Tsia, and B. Jalali. *Serial time-encoded amplified imaging for real-time observation of fast dynamic phenomena.* Nature, 458(7242):1145–1149, apr 2009.

[19] Keisuke Goda, Ali Ayazi, Daniel R Gossett, Jagannath Sadasivam, Cejo K Lonappan, Elodie Sollier, Ali M Fard, Soojung Claire Hur, Jost Adam, Coleman Murray, et al. *High-throughput single-microparticle imaging flow analyzer.* Proceedings of the National Academy of Sciences, 109(29):11630–11635, 2012.

[20] Cheng Lei, Hirofumi Kobayashi, Yi Wu, Ming Li, Akihiro Isozaki, Atsushi Yasumoto, Hideharu Mikami, Takuro Ito, Nao Nitta, Takeaki Sugimura, Makoto Yamada, Yutaka Yatomi, Dino Di Carlo, Yasuyuki Ozeki, and Keisuke Goda. *High-throughput imaging flow cytometry by optofluidic time-stretch microscopy.* Nat. Protoc., 13(7), jul 2018.

[21] Hirofumi Kobayashi, Cheng Lei, Yi Wu, Ailin Mao, Yiyue Jiang, Baoshan Guo, Yasuyuki Ozeki, and Keisuke Goda. *Label-free detection of cellular drug responses by high-throughput bright-field imaging and machine learning.* Scientific reports, 7(1):1–9, 2017.

[22] *Creative Commons Attribution: Attribution 4.0 International.* https://creativecommons.org/licenses/by/4.0/. Accessed: 2021-04-30.

[23] Charis Mesaritakis, Adonis Bogris, Alexandros Kapsalis, and Dimitris Syvridis. *High-speed all-optical pattern recognition of dispersive Fourier images through a photonic reservoir computing subsystem.* Optics letters, 40(14):3416–3419, 2015.

[24] Claire Lifan Chen, Ata Mahjoubfar, Li Chia Tai, Ian K. Blaby, Allen Huang, Kayvan Reza Niazi, and Bahram Jalali. *Deep Learning in Label-free Cell Classification.* Sci. Rep., 6, mar 2016.

[25] Queenie TK Lai, Kelvin CM Lee, Anson HL Tang, Kenneth KY Wong, Hayden KH So, and Kevin K Tsia. *High-throughput time-stretch imaging flow cytometry for multi-class classification of phytoplankton.* Optics Express, 24(25):28170–28184, 2016.

[26] Young Jin Heo, Donghyeon Lee, Junsu Kang, Keondo Lee, and Wan Kyun Chung. *Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip.* Scientific reports, 7(1):1–9, 2017.

[27] Bruno Cornelis, David Blinder, Bart Jansen, Liesbet Lagae, and Peter Schelkens. *Fast and robust Fourier domain-based classification for on-chip lens-free flow cytometry.* Optics Express, 26(11):14329–14339, 2018.

[28] Yuqian Li, Bruno Cornelis, Alexandra Dusa, Geert Vanmeerbeeck, Dries Vercruysse, Erik Sohn, Kamil Blaszkiewicz, Dimiter Prodanov, Peter Schelkens, and Liesbet Lagae. *Accurate label-free 3-part leukocyte recognition with single cell lens-free imaging flow cytometry.* Computers in biology and medicine, 96:147–156, 2018.

[29] Roopam K. Gupta, Mingzhou Chen, Graeme P. A. Malcolm, Nils Hempler, Kishan Dholakia, and Simon J. Powis. *Label-free optical hemogram of granulocytes enhanced by artificial neural networks.* Opt. Express, 27(10):13706, may 2019.

[30] Maxim Lippeveld, Carly Knill, Emma Ladlow, Andrew Fuller, Louise J Michaelis, Yvan Saeys, Andrew Filby, and Daniel Peralta. *Classification of human white blood cells using machine learning for stain-free imaging flow cytometry.* Cytometry Part A, 97(3):308–319, 2020.

[31] Yueqin Li, Ata Mahjoubfar, Claire Lifan Chen, Kayvan Reza Niazi, Li Pei, and Bahram Jalali. *Deep cytometry: deep learning with real-time inference in cell sorting and flow cytometry.* Scientific reports, 9(1):1–12, 2019.

[32] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. *Shortcut learning in deep neural networks.* Nature Machine Intelligence, 2(11):665–673, 2020.

[33] Wenbo Xu, MH Jericho, IA Meinertzhagen, and HJ Kreuzer. *Digital in-line holography for biological applications.* Proceedings of the National Academy of Sciences, 98(20):11301–11305, 2001.

[34] Yichen Wu and Aydogan Ozcan. *Lensless digital holographic microscopy and its applications in biomedicine and environmental monitoring.* Methods, 136:4–16, mar 2018.

[35] Yair Rivenson, Yibo Zhang, Harun Günaydın, Da Teng, and Aydogan Ozcan. *Phase recovery and holographic image reconstruction using deep learning in neural networks.* Light Sci. Appl., 7(2):17141–17141, 2018.

[36] Bendix Schneider, Joni Dambre, and Peter Bienstman. *Fast particle characterization using digital holography and neural networks.* Applied optics, 55(1):133–139, 2016.

1
Machine learning classification

In this chapter we aim to provide a practical and intuitive introduction to the basic machine learning classification concepts and techniques that are referred to in the following chapters. To do so, we demonstrate some of the most relevant aspects by means of a simple numerical example, in which we carry out mock-up classification tasks using a linear classifier. Further details and explanations about most of the discussed content can be found in machine learning introductory textbooks such as [1, 2].

1.1 The classification problem

Machine learning (ML) *classification* is a sub-branch of supervised learning, where each *sample* or *instance* $\mathbf{x}_i$ (different indexes i correspond to different ML samples) belongs to a single class, labelled with y_i. $\mathbf{x}_i$ is usually an array of values (called *features* of $\mathbf{x}_i$) and y_i is a single number (in 2-classes problems) or an array (in multi-class problems), called *label*. The goal is to train an ML algorithm, which learns from a set of labelled *training samples* that are provided to it, to classify (i.e. to assign the correct label to) new samples, which were not employed in the training process. That is, the algorithm is trained with N_{train} tuples ($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), ..., ($\mathbf{x}_{N_{\text{train}}}$, $y_{N_{\text{train}}}$), where N_{train} is the number of samples employed for training. Once trained, the algorithm can calculate from an input sample $\mathbf{x}_i$ a corresponding output *class prediction* $\hat{y}_i$. This latter process, distinguished from the training, is called *inference*. Generally, the aim of ML classification is to successfully train the ML algorithm so that the probability that $\hat{y}_i = y_i$ (i.e. $\mathbf{x}_i$ is correctly classified by the algorithm) is satisfyingly high for $i > N_{\text{train}}$ (i.e. for samples not belonging to the training sample set).

As it is often the case, let us assume that each sample $\mathbf{x}_i$ is a real-valued vector, i.e.:

$$\mathbf{x}_i := [x_{i,1}, x_{i,2}, ..., x_{i,D}]$$

where therefore $x_{i,j}$ is the j^{th} feature of the i^{th} sample $\mathbf{x}_i$ and D is the number of features that represent the considered samples. Note that this means $\mathbf{x}_i$ can be thought of and plotted as a point in a D-dimensional space, called *feature space*, where each coordinate corresponds to a feature. Therefore, a sample set can be represented by a collection of points in the feature space. It should be stressed that, even if the samples in a given sample set have D features, this does not exclude that these samples could still be represented in a subspace of the feature space, i.e. using a smaller number of features, without any significant loss of valuable information. In simpler words, some features could be irrelevant or redundant. For example, in the classification of different types of cars, the number of wheels or the color would be features with low relevance, while the weight of the whole vehicle could be considered as a redundant feature if the weights of the main components are already known. This is a central aspect in machine learning and as such it will be discussed on several occasions in this chapter.

In this chapter we propose a simple numerical experiment as an example to show and explain those general aspects regarding ML classification that will be referred to in the next chapters. Let us imagine that a certain type of measurement is performed on two different objects (or equivalently groups of objects) called A and B, and the outcome of each measurement consists of two quantities, represented by real numbers. Our aim is to carry out a classification task that consists of recognizing which of the two systems were measured, from a single measurement outcome. This problem can be expressed using the mathematical notation we introduced: a sample $\mathbf{x}_i$ represents the i^{th} measurement outcome, while its features $x_{i,1}$ and $x_{i,2}$ represent the two obtained quantities. Moreover, for each sample $\mathbf{x}_i$ we define a suitable label:

$$y_i = \begin{cases} 1 & \text{if } A \text{ was measured} \\ 0 & \text{if } B \text{ was measured} \end{cases}$$

Therefore, for each measurement index i, the information regarding the object of the measurement, that is the *ground truth* of the considered task, is encoded in the corresponding labels. For the sake of simplicity, let us say that $\mathbf{x}_i$ belongs to *class A* if $y_i = 1$, while it belongs to *class B* if y_i = 0. Generally, even considering samples from the same class, different samples may have different feature values, i.e. different coordinate values in the feature space. An intuitive example can be given by a radar that, as an output of its measurement, provides the velocity of an airplane at a certain time. Clearly, even measuring the same airplane model, different velocity values can be obtained at different times. However, it still might be possible to distinguish one airplane model from another on the basis of their velocity, if they have different maximum or minimum velocity.

Generally, the samples belonging to one class can be treated as outcomes of a continuous or discrete multivariate random variable with a certain *probability density function* (PDF, here referred to as *class-specific PDF*) defined on its feature space. Sometimes, in real-life machine learning problems, the distributions at the origin of the available samples are unknown or too complicated to be expressed analytically. In these cases machine learning can provide an important advantage over other approaches, by extracting useful information on PDFs from the available training samples, in order to carry out a given classification task.

Coming back to our numerical experiment, we operationally define two different distributions for the samples generated by the measurements (noiseless for now) on the two objects A and B. That is, the features (where the subscripts A and B indicate the sample class) are generated as follows:

$$\begin{aligned} x_{i,1,A} &= n_i \cos(\theta) - [\sin(n_i\omega) + D/2]\sin(\theta) \\ x_{i,2,A} &= n_i \sin(\theta) + [\sin(n_i\omega) + D/2]\cos(\theta) \\ x_{i,1,B} &= m_i \cos(\theta) - [\sin(m_i\omega) - D/2]\sin(\theta) \\ x_{i,2,B} &= m_i \sin(\theta) + [\sin(m_i\omega) - D/2]\cos(\theta) \end{aligned} \tag{1.1}$$

Here n_i and m_i are real numbers drawn from a 1D *normal (Gaussian) distribution* with standard deviation σ, i.e. from the following PDF:

$$p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n^2}{2\sigma^2}} \tag{1.2}$$

We choose these simple distributions for our samples only because they are convenient for the purpose of demonstrating some basic machine learning aspects. Intuitively, the samples are generated along two sinusoidal curves in the 2D feature space, rotated anticlockwise by an angle θ, with angular frequency ω and at a distance D from one another. In order to provide a visual example, let us plot (in Fig. 1.1 *left*) 100 samples per class with $\sigma = 3$, $\omega = 2$, $\theta = \pi/6$ (rad is the implicitly understood unit), $D = 1.5$, and (in Fig. 1.1 *right*) 1000 samples with $\sigma = 1$, $\omega = 4$, $\theta = 2\pi/3$, $D = 4$.

To solve a classification problem means determining a rule by which each sample can be readily recognized as being part of its class. This can be done by dividing the feature space into different regions, each corresponding to a single class, i.e. by defining a decision boundary (DB). Let us carry out this task manually, i.e. directly looking at the plots and drawing a curve that splits the feature

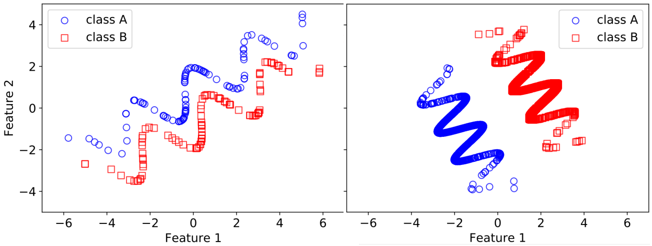


Figure 1.1: Scatter plots of the samples in the feature space, as in the example numerical experiment presented in this chapter. In the two plots different number of samples (100 at the left and 1000 at the right) were considered and different parameters were used to model the class-specific PDFs.

space in two, considering the plotted data (Fig. 1.2). By performing this operation, that from the human perspective appears to be very simple, not only have we carried out the classification task on the available labeled samples, but we have also created a rule to classify unseen and unlabeled samples, assuming that the PDFs remain unchanged. In fact, this is conceptually analogous to what is achieved (or aimed for) by the training of ML classifiers: to learn a general and operational classification rule from a limited number of labelled samples (i.e. the training samples).

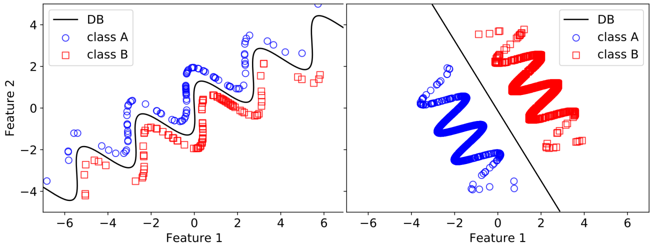


Figure 1.2: Scatter plots of the samples in the feature space. A decision boundary (DB, black curve) was manually drawn to separate the feature space into two regions corresponding to different classes of data.

The ability to classify unseen and unlabeled samples (called *generalization*) is ultimately what makes ML classifiers useful in practice, and it presents some requirements. First of all, it can be achieved only in those regions of the feature space where the two distributions do not significantly overlap. Indeed, in the extreme case where the class-specific PDFs were the same, we may still be able to draw a boundary to separate the points belonging to different classes, since they would not exactly overlap due to randomness. However, the obtained DB would clearly fail to generalize to new samples.

Another key requirement is that the training samples should represent the underlying class-specific PDFs sufficiently well. Generally, more complex and powerful classification models require that the training instances are more varied and numerous, so that they can represent their distribution in greater detail. Let us show this aspect in our example, by performing fewer training measurements (only 10) on the objects A and B, considering the same two combinations of parameters of the previous plotted cases (see Fig. 1.3). The previously drawn DBs are plotted in gray. Looking at the left plot, it can be noticed that just by considering these few training samples, we do not have enough information on the class-specific PDFs anymore, in order to draw a DB that could accurately classify new samples. On the other hand, looking at the right plot, it seems that the same small number of samples is sufficient to guess a good DB. The relevant difference between the two cases is the complexity of the boundary needed to separate the classes. The DB represents the classification model we have learned from the training data, so we can intuitively say that the more complex the required classification model, the harder the classification task, and the higher the number of required training instances. Indeed, this is an important and general rule of thumb in ML, even though it is not always easy or possible to provide a rigorous definition of model complexity, especially when high-dimensional feature spaces are considered. In fact, the difficulty to perform a rigorous and thorough analysis of a problem is often the reason itself why we need to employ ML solutions. Therefore, it should not be surprising if some key ML rules and concepts are expressed in an inexact or heuristic way. Moreover, since the ML foundation

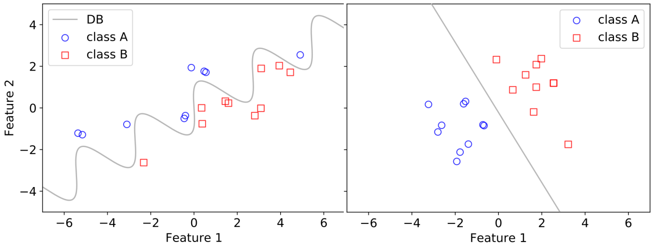


Figure 1.3: Scatter plots of 10 samples per class in the feature space, drawn from the same distributions of the samples plotted in Fig. 1.2. Correct DBs, previously drawn considering many more samples, are represented in gray.

is learning from examples, and humans do the same, often intuition can help to successfully apply ML algorithms.

Nevertheless, in our example, the difference in complexity between the two considered models can be geometrically determined without much ambiguity. In the left plots we employed a sinusoidal (nonlinear) DB, while in the right plots it was sufficient to draw a straight line (a linear DB), which is a simpler separation rule both intuitively and mathematically speaking. Generally, the distinction between linear and nonlinear models is key in ML classification. A linear model solely employs linear combinations of the feature values to be compared to a fixed threshold value (we will discuss the algebraic forms in the next section), which translates in defining one or more linear DBs (i.e. lines in a 2D space, planes in 3D, and generally hyperplanes in a multidimensional space) in the feature space. A nonlinear model is instead based on nonlinear functions of the feature values, and can therefore define nonlinear curves in the feature space. Thus, ML classifiers can be divided into two broad categories: *linear classifiers* and *nonlinear classifiers*. As we will discuss in more detail later on in this chapter, linear classifiers are generally less powerful, but their training is significantly and inherently simpler and they usually require much fewer training samples. Class-specific PDFs that can be separated by a linear model (e.g. in the right-hand side plots) are called *linearly separable*. Clearly, the data plotted in the left-hand side plots are not linearly separable.

In practice, real-life classification tasks are usually much more complicated than the one we depicted, since usually many more features are considered, but also because the presence of noise is an issue. In ML classification we can consider as noise all the variations in feature values that are not correlated with the label values (assuming that the available labels are correct and not noisy, which is not always the case), i.e. the information that is not useful for classification purposes. For example, in the task of distinguishing different car models from a picture, we may have different sources of noise. The noise can be directly generated by the measurement process, e.g. the limited resolution of the picture. Moreover, noise can be introduced because of the environment characteristics, such as fog or the presence of other objects in the picture. But also those characteristics of the car itself that are not relevant to the model type can be seen as noise, e.g. the position or rotation of the car w.r.t. the camera or its colour.

Let us now include noise in our imaginary example: let us assume that the measurements we perform on objects A and B are both affected by additive white Gaussian noise, i.e. every time we obtain a feature value we add to it a random value, drawn from a normal distribution with zero mean and standard deviation σ_N. For simplicity let us only consider the classification task depicted in the left plot of Fig. 1.2, with $D = 3$ and 50 samples per class, for different levels of noise ($\sigma_N =$ 0.3, 0.7, 1, 2 respectively in Fig. 1.4 *a*, *b*, *c*, *d*). As expected, the

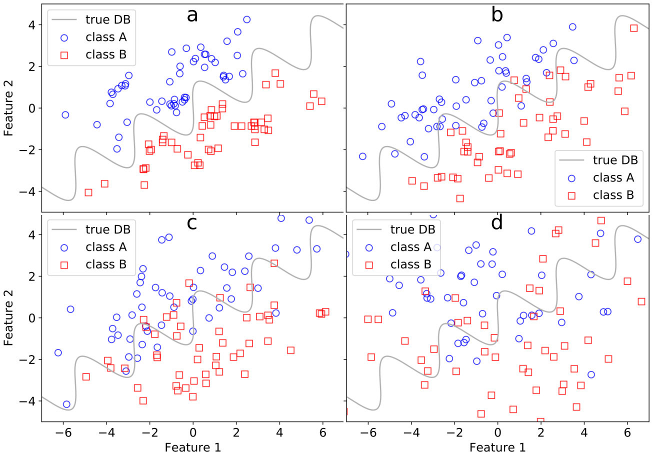


Figure 1.4: Scatter plots of 50 samples per class in the feature space, drawn from the same distributions of the samples plotted in Fig. 1.2, but with $D = 3$ and different levels of noise (σ_N = 0.3, 0.7, 1, 2 respectively in *a*, *b*, *c*, *d*). The correct DB, previously drawn considering noiseless samples, is represented in gray.

addition of noise broadens the effective distributions of the obtained samples, so that the distributions of the two classes end up significantly overlapping for strong enough noise. It should be stressed that the DB drawn considering the left plot in Fig. 1.2, and now represented in gray, is still the optimal boundary for the noisy samples. This is clear because of symmetry reasons, considering that the same isotropic noise is linearly superimposed on the original noiseless distributions and that the same number of samples is considered for both classes. Therefore, we can see that for high noise levels, some classification error is inevitable. For example, by counting the number of misclassified samples in Fig. 1.4 *b* (9 over 100), we can say that the estimated *classification error* (or *error rate*) of the employed model is 9%, or equivalently, the estimated *classification accuracy* is 91%. Furthermore, if we tried to draw a DB just by looking at the plotted noisy samples (i.e. by considering them as training samples), we would draw a boundary that is more and more different from the best one, as the noise level increases. Therefore, noise can affect the classification performance at least in two ways: by directly increasing the overlapping of the class-specific PDFs introducing inevitable errors and, on top of this, by disrupting the training process. Thus, given a certain task, a good ML classifier should present a training process that is sufficiently robust to noise. Moreover, it should be taken into consideration that in many real-life problems there might be a minimum classification error value that cannot be further decreased, no matter the effort spent on the ML algorithm.

It should be noticed that, as opposed to our example, generally the best possible DB can depend on the noise, e.g. when it is more intense for one class than for another or when it is more important not to misclassify the samples belonging to one class w.r.t. others.

1.2 Linear discrimination with logistic regression

In the previous section we have introduced some basic definitions and concepts regarding ML classification, but we have not gone into the details of any specific approach or algorithm. We do this in this section, focusing on linear classifiers and in particular on the *logistic regression* model, which is the one considered in all the following chapters.

We have seen that a solution for a two-class problem can be represented by a suitable DB in the feature space. In particular, linear classifiers can learn linear DBs, which separate the feature space into two regions that can be mathematically expressed by the inequality:

$$g(\mathbf{x}_i|\mathbf{w}, w_0) := \mathbf{w} \cdot \mathbf{x}_i + w_0 > 0 \tag{1.3}$$

Where $\cdot$ is the scalar product in the feature space, $\mathbf{w}$ is a vector whose elements w_j are called *synaptic weights* (often just *weights*) and w_0 is a scalar called *intercept*. The function g is called *linear discriminant* and consists of a *linear combination* (or *weighted sum*) of the features of the sample $\mathbf{x}_i$ added to the intercept. Clearly, the DB is defined by substituting $>$ with $=$ in the inequality. Therefore, given certain weights and intercept (that are learned through a training algorithm), we can assign a class prediction to a sample $\mathbf{x}_i$ as follows:

$$\hat{y}_i = \begin{cases} 1 & \text{if } g(\mathbf{x}_i|\mathbf{w}, w_0) > 0 \\ 0 & \text{if } g(\mathbf{x}_i|\mathbf{w}, w_0) \leq 0 \end{cases} \tag{1.4}$$

It can be noticed that the defined classification model corresponds to comparing a linear combination with a threshold value $-w_0$, by moving the intercept to the right hand side in equation 1.3.

Let us now see on what basis logistic regression calculates weights and intercept in order to correctly classify the training samples. To do so, we first introduce a few notations and concepts taken from probability theory. A probability estimation that is particularly relevant in classification is the conditional probability $P(C|\mathbf{x}_i)$, which is the probability that a sample $\mathbf{x}_i$ belongs to a class C. Indeed, if we can estimate such a probability for each considered class, we can simply classify $\mathbf{x}_i$ as belonging to the class with the highest probability value in order to maximize our classification accuracy over many samples. Let us notice that in a two-class problem (with classes A and B) we have $P(C = A|\mathbf{x}_i) + P(C = B|\mathbf{x}_i) = 1$. Moreover, with $P(\mathbf{x}_i|C)$ we indicate the conditional probability that a sample belonging to a class C is equal to $\mathbf{x}_i$, which corresponds to the previously introduced class-specific PDF. $P(\mathbf{x}_i)$ is the marginal probability that the sample $\mathbf{x}_i$ is obtained regardless of the class, and $P(C)$ is the marginal probability that any sample from a class C is obtained, regardless of the sample. $P(C)$ is called *prior* probability, since it contains the information we have before looking at any sample. These four probabilities are linked by the well-known Bayes' rule:

$$P(C|\mathbf{x}_i) = \frac{P(C)P(\mathbf{x}_i|C)}{P(\mathbf{x}_i)}$$

In logistic regression we model the logarithm of the ratio of class-specific PDFs in the feature space with a linear relation, or equivalently in the two-class case we assume:

$$\log\left(\frac{P(\mathbf{x}_i|A)}{P(\mathbf{x}_i|B)}\right) = \mathbf{w} \cdot \mathbf{x}_i + \widetilde{w}_0 \tag{1.5}$$

Here $\widetilde{w}_0$ is a real number. It should be stressed that it is not important if this model does not accurately describe the actual class-specific PDFs in our classification problem, but it is sufficient that it provides a good enough approximation for the only sake of separating two classes with a linear DB. Moreover, we can notice that in order to find a good linear DB, we aim to know the ratio between $P(A|\mathbf{x}_i)/P(B|\mathbf{x}_i)$ rather the one in Eq. 1.5. The latter, however, can be more directly estimated from the training samples. Indeed, as we mentioned, the average classification accuracy can be maximized by classifying $\mathbf{x}_i$ as belonging to class A if such a ratio is greater than 1, as belonging to class B otherwise. In light of this we can see that the following inequality is equivalent to the linear discrimination model expressed by Relations 1.3 and 1.4:

$$\log\left(\frac{P(A|\mathbf{x}_i)}{P(B|\mathbf{x}_i)}\right) = \mathbf{w} \cdot \mathbf{x}_i + w_0 > 0 \tag{1.6}$$

Bayes' rule provides the link between this equation and Eq. 1.5:

$$\log\left(\frac{P(A|\mathbf{x}_i)}{P(B|\mathbf{x}_i)}\right) = \log\left(\frac{P(\mathbf{x}_i|A)}{P(\mathbf{x}_i|B)}\right) + \log\left(\frac{P(A)}{P(B)}\right)$$
$$\text{with} \quad w_0 = \widetilde{w}_0 + \log\left(\frac{P(A)}{P(B)}\right)$$

By rearranging Eq. 1.6, remembering that $P(B|\mathbf{x}_i) = 1 - P(A|\mathbf{x}_i)$, we can find how the conditional probability that $\mathbf{x}_i$ belongs to class A is modelled in logistic regression:

$$P(A|\mathbf{x}_i) = l(g(\mathbf{x}_i|\mathbf{w}, w_0)) = \frac{1}{1 + \exp(-g(\mathbf{x}_i|\mathbf{w}, w_0))} \tag{1.7}$$

where l is called *logistic* (or *sigmoid*) function. It should be noticed that Eq. 1.7 represents a smooth version of the step-like function $\hat{y}_i(g(\mathbf{x}_i|\mathbf{w}, w_0))$ defined by Eq. 1.4, which gives a prediction of the classification label of a sample $\mathbf{x}_i$. Therefore, we can see that logistic regression, additionally to label prediction, also provides an estimation of the probability that a predicted label is correct, i.e. the risk of misclassification is quantified for each sample.

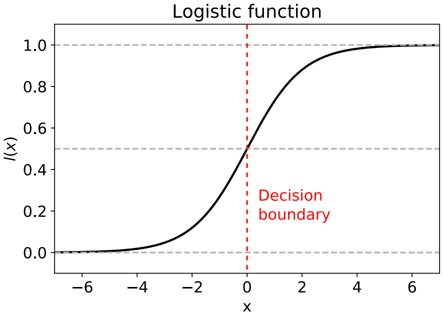


Figure 1.5: Logistic function (also called sigmoid). The function presents two asymptotes y = 0 and y = 1, and we have $l(0) = 0.5$. For example, in a two-class problem a decision boundary can be set at $x = 0$, corresponding to the points in the feature space that have the same estimated probability of belonging to either class.

This information can be useful in different cases, for instance when we tolerate less the misclassification of samples belonging to a class w.r.t. another. To present a practical example, let us imagine that our samples are microscope images of biological tissue taken from patients and that we aim to automatically detect cancer formation, so that in case of detection we can proceed with more accurate analysis. In this case, we probably want to miss as few cases of cancer as possible, even if we might sometimes wrongly detect it. Then, with reference to Fig. 1.5, which would represent the estimated probability that a sample presents cancer formation, we could just move the decision boundary towards the left, i.e. setting a threshold lower than 0 for our linear discriminant, thus lowering the probability of misclassifying cancer samples. In this and in the next chapters, however, we always consider the decision threshold at $P(C|\mathbf{x}_i) = 0.5$ in two-class problems.

We have seen how logistic regression models a classification problem, but we still need to understand how this model can be optimized by a training algorithm, i.e. how optimal values of the model parameters **w** and w_0 are calculated exploiting the training samples. Let us consider a set of training samples and ground truth labels pairs $\mathcal{X} = \{\mathbf{x}_i, y_i\}$, and let us assume that y_i, given $\mathbf{x}_i$, is a random variable with probability of being 1 and 0 given respectively by $p_i \equiv P(A|\mathbf{x}_i)$ (as in Eq. 1.7) and $P(B|\mathbf{x}_i) = 1 - p_i$. In other words, y_i has a Bernoulli distribution and we can simply write $P(y_i|\mathbf{x}_i) = p_i^{y_i}(1-p_i)^{1-y_i}$. Assuming that the samples are independently generated, we can then calculate the probability of sampling the whole training set from the considered distribution, which in turn is parametrized by **w** and w_0:

$$P(\mathcal{X}|\mathbf{w}, w_0) = \prod_i P(y_i|\mathbf{x}_i) = \prod_i p_i^{y_i}(1-p_i)^{1-y_i}$$

Therefore, by maximizing this quantity through optimization of the model parameters **w** and w_0, we can obtain the parameter combination that best explains the given training set, assuming that they are distributed as indicated in Eq 1.7. Most importantly, even without such an assumption, in a general 2-class classification problem this corresponds to finding a good linear discriminant that separates the two classes through a linear DB. Of course, a completely successful classification of the training samples is guaranteed only if the classes are linearly separable.

The maximization of $P(\mathcal{X}|\mathbf{w}, w_0)$ is equivalent to the minimization of:

$$\begin{aligned} E(\mathbf{w}, w_0|\mathcal{X}) &= -\log(P(\mathcal{X}|\mathbf{w}, w_0)) \\ &= -\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i) \end{aligned} \tag{1.8}$$

This is easier to treat mathematically. This particular expression of function E, which is the *cost function* in logistic regression, is called *cross-entropy*. We can notice that each training sample $\mathbf{x}_i$ contributes to the cost function with a term $-\log(p_i)$ if $y_i = 1$ and with a term $-\log(1-p_i)$ if $y_i = 0$ (see plot in Fig. 1.6). Thus, the cost related to a single sample raises faster than linearly as the probability of misclassification increases. Because of the nonlinearity of p_i as a function of the parameters, the cost function of logistic regression is usually minimized iteratively by means of *gradient descent*, which is a very popular optimization algorithm in machine learning. In practice, random small values (e.g. in the interval $[-0.01, 0.01]$) are initially assigned to the the parameters, then the gradient of the error function calculated for the current parameter values is computed. The parameters are updated by subtracting to them a term proportional

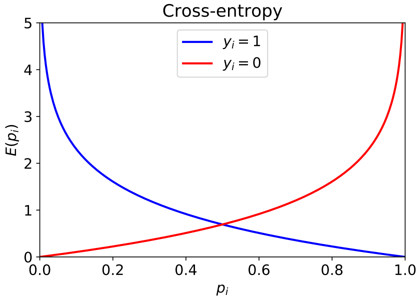


Figure 1.6: Contribution to the cost function (cross-entropy) in the two cases where the class label is 1 (blue) and 0 (red).

to the computed gradient, i.e. at each training iteration indexed by t:

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t + \Delta\mathbf{w}_t \\ w_{0,t+1} &= w_{0,t} + \Delta w_{0,t} \\ \Delta\mathbf{w}_t &= -\eta \left.\frac{\partial E}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_t} \\ \Delta w_{0,t} &= -\eta \left.\frac{\partial E}{\partial w_0}\right|_{w=w_{0,t}} \end{aligned} \tag{1.9}$$

Here η is called *learning rate*, and regulates how big the steps are that are taken by the parameters along the direction of the cost function gradient at each training iteration. In the particular case of logistic regression, the weight updates are:

$$\begin{aligned} \Delta\mathbf{w}_t &= \eta \sum_i (y_i - p_i)\mathbf{x}_i \\ \Delta w_{0,t} &= \eta \sum_i (y_i - p_i) \end{aligned} \tag{1.10}$$

The training iterations are usually repeated until convergence, i.e. when the parameter updates become smaller than a fixed tolerance threshold. When the parameters reach values corresponding to a minimum of the cost function, the algorithm converges if the learning rate is not overly large. Once this happens, we can consider our classifier trained. If the training is successful, the obtained classification model will generalize satisfyingly well also on new unseen samples. It should be stressed that the general necessary conditions for this to happen, are that the classifier is powerful enough to learn a suitable model to separate the classes in the given feature space, and that the training samples represent the general sample distribution well enough. In the next section we will demonstrate and discuss these aspects through practical examples.

Logistic regression can be easily generalized to problems with more than two classes, becoming a *multinomial* logistic regression. In this case, an optimal set of weights and intercept is found for each class. For inference, the corresponding weighted sums are calculated, and the one with the largest outcome indicates the predicted class. However, multi-class problems can be also tackled by training multiple traditional two-class logistic regressions and by combining the corresponding outcomes for inference. For example, in the *one versus all* approach, for each class we train a logistic regression considering the problem of discriminating that class from the union of all the others. As with multinomial logistic regression, inference is performed by considering the set of weights and intercept whose application to the given sample provides the largest outcome. A popular way to visualize the performance of a classification algorithm applied to a multi-class problem is by displaying a *confusion matrix*, or *error matrix*. This is a specific table layout, where each row represents the samples belonging to a class and each column represents the samples classified by the algorithm as belonging to a class. In particular, the element i, j of the matrix corresponds to the fraction (or the number) of the samples belonging to the i^{th} class, that were classified as belonging to the j^{th} class. Therefore, if the performed classification is flawless, only the diagonal terms (corresponding to the class-specific accuracies) are non-zero. Importantly, a confusion matrix allows to quickly visualize to what degree each pair of classes are confused by the classifier.

Finally, we can graphically represent a linear classifier model obtained through logistic regression by drawing the corresponding *neural network architecture diagram* (Fig. 1.7).

1.3 Underfitting, overfitting and validation

Let us now apply logistic regression to the numerical example we introduced in Section 1.1. We can see that if the classes are linearly separable, a correct DB is found (Fig. 1.8 *left*). Otherwise, logistic regression can still be useful, since it provides the linear DB that minimizes the cross-entropy, which is usually close to the linear separation with the highest possible accuracy (Fig. 1.8 *right*). Yet, in this case a more powerful nonlinear classifier could clearly define a better DB, reaching a significantly higher classification accuracy. Generally, the inability of a classifier to perform a successful classification because it can only learn too simple models, is called *underfitting*. Its counterpart, *overfitting*, occurs instead when the training samples are not enough to represent the underlying distributions at the level of detail required to properly learn a complex and powerful

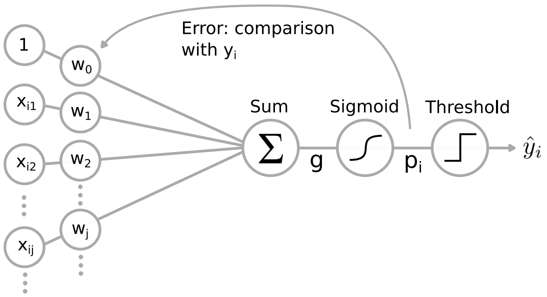


Figure 1.7: Neural network architecture diagram corresponding to a linear classifier (2 classes) obtained through logistic regression. The features of a sample $\mathbf{x}_i$ are multiplied by the weights in $\mathbf{w}$ and summed together with the intercept w_0. The resulting linear discriminant is fed into the sigmoid function, whose output is compared to a threshold in order to obtain the class label estimation. The network is trained by small iterative changes in weights, determined by the cross-entropy cost function on the whole training set.

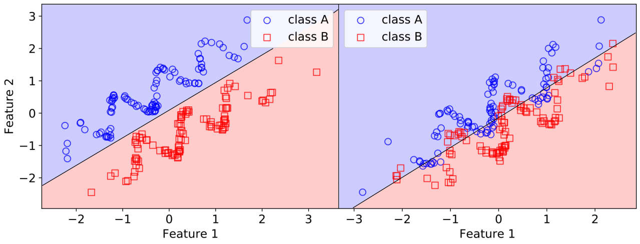


Figure 1.8: Classification with logistic regression. The two regions, corresponding to the prediction of the classes A and B, defined by the decision boundary are respectively coloured in light blue and light red. In the *left* plot the classes are linearly separable, while in the *right* plot they are not.

model. Overfitting is a very common issue in machine learning and it is worth discussing by means of dedicated examples. However, the simple linear classifier that we are considering is not suitable to demonstrate overfitting. So let us first see how we can make it more powerful.

Until now we have considered relatively simple classification tasks and therefore we still have not shown how linear classifiers can be powerful and useful in real-life applications. Indeed, all the classification problems we have seen could be easily solved by humans, once the samples are visualized in the feature space. Of course, logistic regression can solve a large number of such simple problems in a very short time even using a simple laptop, and this is already an advantage w.r.t. human labour. But more importantly, logistic regression can easily be employed to carry out high-dimensional classification tasks (i.e. with a number of features > 3, or even $\gg 3$), which cannot be represented by plotting the samples in the feature space. These high-dimensional problems can be exceedingly difficult to solve without machine learning. Coming back to our numerical example, we may obtain higher dimensional samples by measuring more characteristics of the objects A and B, thus obtaining from each measurement not only the features $x_{i,1}$ and $x_{i,2}$, but also $x_{i,3}$, $x_{i,4}$, etc. In this way, if the features are relevant and not redundant, richer information is obtained and the classification might become easier.

Another important and related aspect is that linear classifiers can be made more powerful by *feature expansion*. This consists of increasing the number of employed features without modifying the measurements, but just calculating for each sample new features from the initial ones. Feature expansion is key in many ML applications. For example, since we know what the distribution of classes A and B looks like (see Section 1.1), we can guess that additional features obtained by suitable sinusoidal functions of the initial features $x_{i,1}$ and $x_{i,2}$ can be of help when the samples are not linearly separable, e.g. as in the task represented in Fig. 1.8 *right*. Let us try this out by adding the following new features to the $\mathbf{x}_i$ vectors: $x_{i,3} := \sin(x_{i,1}\omega)$ and $x_{i,4} := \sin(x_{i,2}\omega)$. This way, the logistic regression will draw a linear DB in the 4D feature space, which becomes a nonlinear DB once projected back to the original 2D feature space (Fig 1.9 *a, c*). We can notice that the new DB can better separate the classes than a linear one. Generally, feature expansion can be either considered as part of the classifier or not. Indeed, in our case, we could say that we employed a linear classifier on the 4 features $x_{i,1}$, $x_{i,2}$, $x_{i,3}$ and $x_{i,4}$, where the latter two features were calculated from the original ones. Or equivalently, we could state that we classified the original two features with a nonlinear classifier, which comprises the feature expansion operations in its algorithm.

In order to check if the learned classification model can generalize to unseen samples, we sample additional instances from both classes (i.e. we take additional measurements on objects A and B), obtaining a *test set* $\mathcal{X}^{test}$ (plotted in Fig. 1.8 *b, d*). The classification accuracy (called *test accuracy* as opposed to *training accuracy*) obtained by applying the learned model on the test samples is crucial to *validate* the performance of a ML classifier, i.e. to evaluate its generalization capability.

Overfitting is often the main cause of bad generalization: in such a case, a classifier is trained on samples that do not adequately represent their distribu-

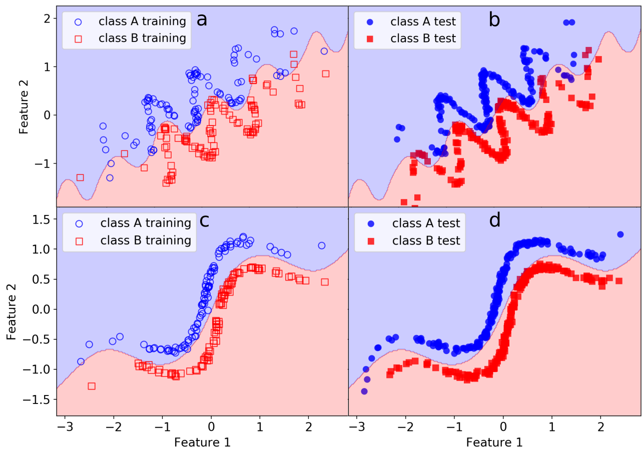


Figure 1.9: Classification with logistic regression, adding two new features $x_{i,3} := \sin(x_{i,1}\omega)$ and $x_{i,4} := \sin(x_{i,2}\omega)$ to the original ones through feature expansion. The linear DB in the 4D feature space was projected in the original 2D feature space. Thanks to feature expansion, a more accurate DB was obtained. *a, c*: two cases (using different distribution parameters) of training samples used to learn the DB. *b, d*: corresponding test samples, plotted to check how the learned model generalizes to unseen samples.

tion. For example, let us consider the same classification task with the two additional features (as in Fig. 1.9 *a*), but this time we employ only 10 training samples per class instead of 100. We notice that the learned DB can perfectly classify the training samples (Fig. 1.10 *a, c*) but clearly fails to generalize to the unseen test samples (Fig. 1.10 *b, d*). Moreover, it should be stressed that the training samples in the two plots were sampled from the same distribution, but because of their low number, very different classification models were learned. Indeed, not only does overfitting degrade the test accuracy, but also increases its *variance*.

A crucial aspect of overfitting is that the severity of its negative effects depends both on the number of training samples and on the complexity of the learnable models. In general, the more powerful a ML classifier is in fitting sample distributions, the higher the required number of training samples in order to avoid (or just to keep under control) overfitting. Let us exemplify this by training our linear classifier on the same low number of samples per class, but without feature expansion. We notice that the training of the linear model is much less affected by overfitting and it provides a better test accuracy with relatively low

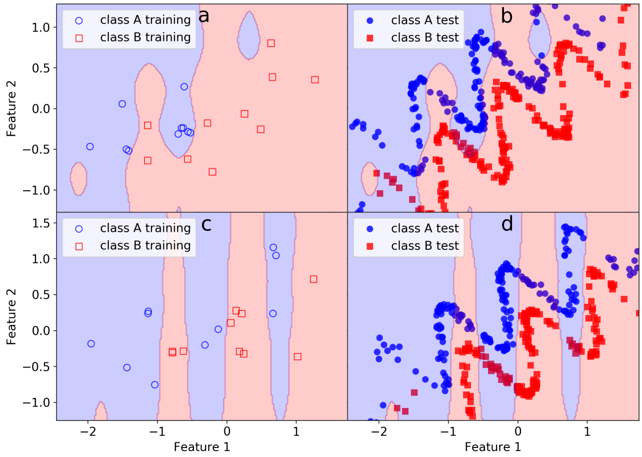


Figure 1.10: Classification with logistic regression, employing the features: $x_{i,1}$, $x_{i,2}$, $x_{i,3} := \sin(x_{i,1}\omega)$ and $x_{i,4} := \sin(x_{i,2}\omega)$. *a, c*: only 10 training instances per class are considered in each case, drawn from the same probability distribution as in Fig. 1.9 *a, b*. Because of overfitting, the learned model perfectly classifies the training samples but fails to generalize to unseen test samples, plotted in *b* and *d*.

variance (Fig. 1.11).

1.4 Learning curve, cross-validation and hyperparameters tuning

In many practical cases, the available number of labeled samples that we can use to train a classifier is not enough to completely avoid overfitting, e.g. because generating or correctly labeling samples is expensive, or because we use a very powerful model that easily overfits the data and it becomes impractical to avoid overfitting at all costs. Therefore, often we need to cope with overfitting and try to achieve a satisfying result by efficiently employing a limited number of samples. In these cases it is important to precisely evaluate the classification performance for different numbers of employed training samples. The most popular way to do this is to draw a *learning curve*, i.e. the classification error as a function of the number of training samples. We provide here two examples of learning curves (Fig. 1.12 *a, b*) applied respectively to the discussed cases

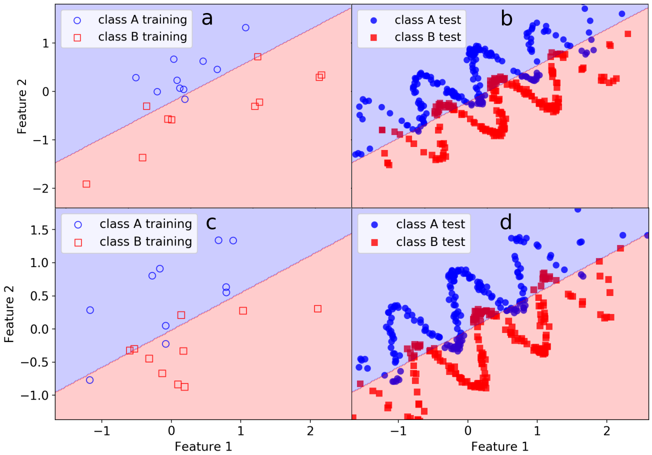


Figure 1.11: Classification with logistic regression, employing the original features $x_{i,1}$, $x_{i,2}$. *a, c*: only 10 training instances per class are considered in each case, drawn from the same probability distribution as in Fig. 1.9 *a, b*. Even though a very low number of training samples are used, the learned model was simple enough not to be significantly affected by overfitting and could still generalize to unseen test samples, plotted in *b* and *d*.

with and without sinusoidal feature expansion, corresponding to Fig. 1.10 and Fig. 1.11. In particular, for each point of the curve, we randomly generated 10 different training sets with the same number of samples and we evaluated the corresponding training error rates. Moreover, we evaluated the test error rate of each trained model by applying it to the same test set of 200 samples per class. We plotted the mean error rate value at each point and the corresponding error bars (twice the standard deviation). Thus, for each number of training samples, it is easy to check how overfitting affects the classification, both in terms of the discrepancy between training and test error rates and in terms of variance. We can notice that the more complex and powerful model (Fig. 1.12 *a*) provides a lower classification error than the simpler model (Fig. 1.12 *b*), but it requires more training samples to avoid overfitting. It should be stressed that it is usually important to estimate the variance of the classification performance, in order to obtain a general evaluation of how good a classifier is at learning a given task.

However, when the production of labeled samples is costly, estimating the classification performance as we just did is not efficient. Indeed, we employed a large number of test samples that could have been used for training to better

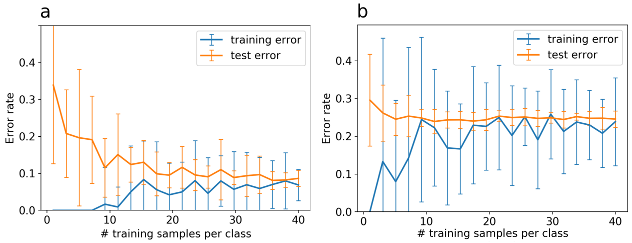


Figure 1.12: Learning curves (classification error rate v.s. number of training samples) corresponding to the classification respectively with (*a*) and without (*b*) sinusoidal feature expansion, represented in Fig. 1.10 and Fig. 1.11. The plotted points and the error bars represent respectively the mean and the confidence interval (twice the standard deviation) corresponding to 10 different randomly generated training sets with the same number of samples. We can notice that the more complex model (*a*) provides a lower classification error than the simpler model (*b*), but it requires more training samples to avoid overfitting.

learn the classification. On the other hand, it is important to have enough test samples to correctly estimate the general classification performance. Moreover, in order to estimate the error variance at each point of the learning curve, we generated each time new training sets, discarding the used ones instead of reusing them. A popular way to efficiently employ all the available samples to train a ML model and estimate test accuracy and variance, is *k-fold cross-validation*, whose functioning is schematized in Fig. 1.13. In this specific type of validation

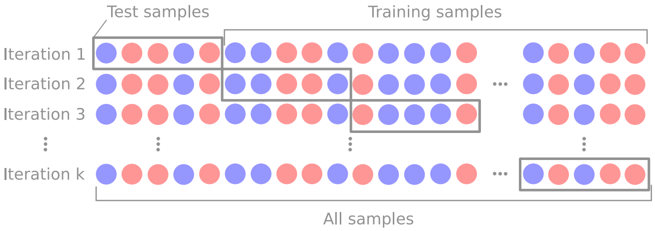


Figure 1.13: Diagram representing how the available samples are efficiently employed to train and test a ML model using k-fold cross-validation. Blue and red circles represent samples from two different classes.

technique, we train and validate a classifier at each one of k iterations, splitting the available samples into two disjoint subsets (training set and *validation set*), so that each iteration employs disjoint validation sets with (approximately) the same cardinality. Therefore, k performance estimations are obtained using different validation sets and slightly different training sets. The main advantage of such an approach is that we can, up to a limited extent, overcome the scarcity of samples in exchange for computational power. For example, let us assume that we only have 100 samples and we would like to employ the maximum possible number of training samples to reduce overfitting, but we would also want to employ the maximum possible number of test samples in order to obtain a good enough performance estimation. In an extreme case, we can employ a 100-fold cross-validation, allowing us to employ 99 samples for training, while testing our model on all the available samples one by one, at the cost of performing the training 100 times.

Another important use for cross-validation is hyperparameter tuning. *Hyperparameters* are all the parameters of a ML pipeline that can influence (and thus control) the training process, as opposed to the model parameters that are automatically optimized by the training, such as weights and intercept in logistic regression. To provide an example, when we employed feature expansion to enhance the classification power of our classifier (Fig. 1.9), we calculated new features from the original ones, through a sinusoidal function with angular frequency ω. We chose the same frequency that was used to generate the original features (Eq. 1.1), since we knew that it was the one that characterized the sample distribution. Instead, let us assume, more realistically, that we do not have this information, but we just believe that the sample distribution has a sinusoidal shape (we generally call this kind of information a *prior*, since it is known before looking at the instances). In this case, we may add the features $x_{i,3} := \sin(x_{i,1}\tilde{\omega})$ and $x_{i,4} := \sin(x_{i,2}\tilde{\omega})$, where $\tilde{\omega}$ is a hyperparameter to optimize. In practice, we can repeat feature expansion many times for different values of $\tilde{\omega}$, including in the final classifier only the one that provides the best classification performance. To do so, we need to test each classifier version corresponding to a different $\tilde{\omega}$ value. However, to do this, we cannot use the test set, since a crucial condition for a valid evaluation of the final classifier is that the test samples do not influence the training process in any way. Instead, we can split the sample set into three subsets: a training set, a *validation set* and a test set. The validation set is then used, similarly to a test set, to evaluate the classifier performance for fixed values of the hyperparamters. Once all hyperparameters combinations that are to be explored have been evaluated on the validation set, the best classifier version is chosen and it is eventually tested on the test set. In these cases, k-fold cross-validation is usually employed to create many versions of the training set and validation set out of the available samples (excluding the test set, which must be left unseen until the end), for efficient hyperparameter selection with a limited number of samples. If the computational resources allow it, a k-fold cross-validation cycle for hyperparameters selection can be *nested* into an external one, which is in turn employed to increase the number of samples used for the final test. It should be stressed that hyperparameter tuning, as a net result, increases the classification power of a ML application and as such it also increases the risk of overfitting.

1.5 Dimensionality reduction and regularization

We have seen that through feature expansion the sample dimensionality can be expanded to make a classifier more powerful. However, we have also seen that this can substantially increase the risk of overfitting, requiring more samples to successfully train the ML model. Moreover, the computational cost of training usually increases at least linearly with the sample dimensionality and the number of training samples. Therefore, employing many features might require an exceedingly long training time. For these reasons, it is often necessary to reduce the number of features (this process is called *dimensionality reduction*) by either selecting only the most relevant and less redundant ones (i.e. performing a *feature selection*) or by calculating a fewer number of better features from the discarded original ones (which is called *feature extraction*).

Let us now provide a numerical example of how employing irrelevant features might disrupt the training process due to overfitting. First of all, we show that employing values of $\tilde{\omega}$ that are significantly different from the angular frequency $\omega = 6$ used to define the sample distribution provides not so relevant features. Indeed, we can see that adding the features obtained with both $\tilde{\omega} = 3\omega$ and $\tilde{\omega} = \omega/3$ provide no evident advantage over the original linear classification with only $x_{i,1}$ and $x_{i,2}$ (Fig. 1.14 *a* and *b* respectively). But let us see what happens if we employ many values of $\tilde{\omega}$ at the same time, including the relevant one ω. In particular, we add 22 sinusoidal features corresponding to $\tilde{\omega}$ = 0.6, 0.95, ..., ω, ..., 37.86, 60 (i.e. the values are equidistant in a logarithmic scale). In this case, the logistic regression could potentially employ only the relevant features corresponding to $\tilde{\omega} = \omega$ while ignoring the others (by assigning them null weights). However, the model complexity has increased so much that, because of overfitting, a noisy and non-general DB is generated, even using 50 samples per class (Fig 1.14 *c*). Indeed, comparing the corresponding learning curve (Fig. 1.15 *a*) with the one obtained with $\tilde{\omega} = \omega$ (Fig. 1.12 *a*), we notice that a much larger number of training samples is needed to overcome overfitting effects. However, when enough training samples are employed, the combination of several feature that alone were not very useful, allowed to substantially decrease the classification error. We will further discuss this effect in Section 1.7.

In this example we considered the case where in feature expansion we might have added too many features, and therefore dimensionality reduction can be

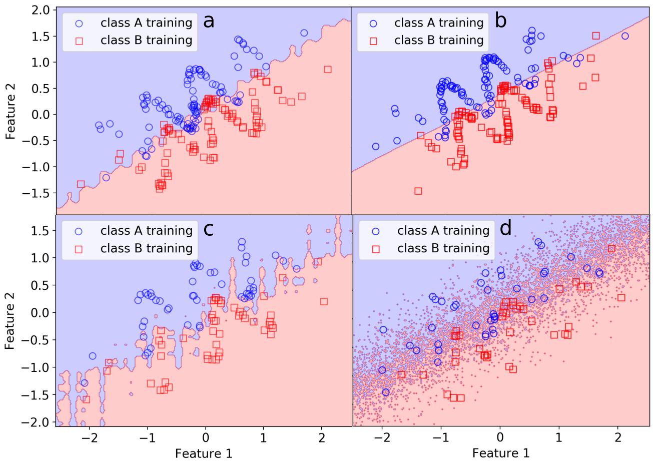


Figure 1.14: Classification with logistic regression. *a, b*: irrelevant features (corresponding to $\tilde{\omega} = 3\omega$ and $\tilde{\omega} = \omega/3$ respectively) were added to the original ones. *c*: 22 sinusoidal features corresponding to $\tilde{\omega}$ = 0.6, 0.95, ..., ω, ..., 37.86, 60 (i.e. with values equidistant in a logarithmic scale) were added. *d*: 22 noisy features were added, sampled form a normal distribution with standard deviation of 10.

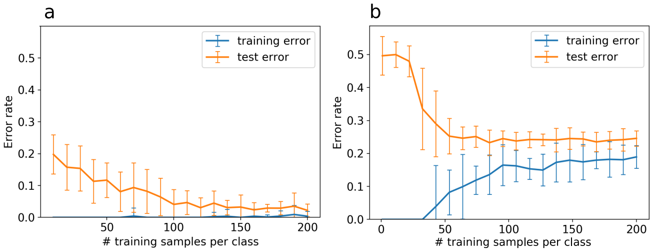


Figure 1.15: *a, b*: Learning curves corresponding respectively to the cases where 22 sinusoidal features (Fig. 1.14 *c*) and 22 noisy features (Fig. 1.14 *d*) were added to the original features $x_{i,1}$ and $x_{i,2}$.

helpful in decreasing overfitting. Often, however, irrelevant or redundant features are not the product of feature expansion, but are directly and intrinsically generated by sample measurement. For instance, if we want to classify images on the basis of their content, every pixel value is a feature of a sample image. In such a case, a large number of pixels are likely to be just irrelevant or redundant for the sake of classification. To exemplify this case, let us add to the original features 22 noisy features, sampled form a normal distribution with standard deviation of 10. Thus, these features do not contain any class-related information and are therefore useless by definition. Still, we can see that these have a negative effect on the learning process (Fig. 1.14 *d*), which requires significantly more samples in order to reduce overfitting (Fig. 1.15 *b*).

Among several approaches and techniques for dimensionality reduction, here we just briefly discuss few basic and popular ones. Let us start with *principal component analysis* (PCA), which is an unsupervised method, since it does not use the class label information. PCA aims to extract the features that most explain the sample variance in a non-redundant way, by means of scalar projections (i.e. linear combinations) of the initial features on a suitable orthonormal basis. In particular, it first finds a direction (called first principal component) in the initial feature space so that the scalar projection of the samples on such a direction has the highest possible variance. Thus, a first feature is extracted through such a projection. Afterwards, it repeats the same operation with the constraint that the second direction with maximum variance (called second principal component) must be orthogonal to the first one, and so on. By employing only a limited number of the extracted features with the highest variance, it is then possible to reduce the sample dimensionality with the least possible loss of variance allowed by linear transformations. Therefore, PCA can be particularly useful when the class-related information is one of the factors that originates most of the sample variance, and when there are many redundant or irrelevant and low-variance features.

However, it is possible that the most of the sample variance is not relevant for the classification purpose. For instance, in the left plot of Fig. 1.8 we can see that the direction perpendicular to the DB is the one that best separates the sample but represents the lowest sample variance. In these cases, PCA is clearly not the best choice, and instead it might be beneficial to employ a supervised dimensionality reduction method that takes the class labels into consideration, such as *linear discriminant analysis* (LDA). LDA is similar to PCA in that it aims to find the most suitable scalar projections of the initial feature space to extract few relevant and non-redundant features. However, the orthogonal directions considered for projections are chosen such that the linear separation between the classes is maximized, under few assumptions among which multivariate normality of the class-specific PDFs. For example, considering again the case depicted in the left plot of Fig. 1.8, LDA would choose for its first projection the direction perpendicular to the DB. In this way, the samples would be linearly separable with only that one extracted feature, while initially both the original features were required for linear separability. Intuitively, LDA can rotate the samples in the feature space so that class separation is expressed on fewer orthogonal axes.

We have just seen that feature extraction can be used for dimensionality reduction by calculating fewer improved features from the initial ones. Nevertheless also this operation increases the complexity of the employed ML pipeline, so it is important to ensure that the corresponding increase in overfitting risk is overcome by the overfitting decrease due to the reduction of the features employed by the subsequent classifier. Moreover, to calculate the extracted features, these methods still employ all the initial features even though some of them might be useless. Therefore, in some cases feature selection is preferable, which allows to completely remove unimportant features from the classification pipeline. This can be particularly important in those cases where each initial feature corresponds to a value whose measurement comes with a considerable cost. For example, to predict the occurrence of medical conditions in order to prevent them, such as heart attack or seizure, we might first try to apply many sensors to the human body to understand which measurements are the most useful for a given ML classification. Therefore, in this case the selection of fewer good features directly translates into a reduction in the number of required sensors.

Another aspect to consider is that the feature extraction operation, similarly to expansion, has a computational cost that has to be paid also in classification inference, while the computational cost of feature selection needs to be paid only in the development of a ML model. For example, if we want to carry out fast image classification, we might have something like 1 million pixel values for each sample. Assuming that there are a relatively small number of pixels that are significantly more relevant than the others, feature extraction would still require several linear combinations on all the initial pixels, while this can be avoided using feature selection instead, which might therefore substantially enhance the computational efficiency of inference. A popular example is given by *region of interest* (ROI) extraction, where only specific areas in an image are selected for further analysis.

Computationally efficient feature selection can be performed through *filter-based* methods, which rely on the calculation of computationally cheap statistics in order to select a feature subset. Usually, these statistics provide a heuristic quantitative evaluation of the relevance and/or redundancy of the initial features, allowing to rank them accordingly. For instance, the correlation between pairs of features can be calculated to estimate the redundancy of the conveyed information, while the correlation between the features and the class labels can provide a relevance measure.

A more powerful but computationally expensive feature selection approach is given by *wrappers*, which are algorithms that directly test the performance of a specific ML model when trained on subsets formed by different combinations of the initial features. The subset that scores best is chosen. For example, in the *for-*

ward selection algorithm, we start with a void feature subset and at each iteration we evaluate the performance (e.g. the validation accuracy) of our classifier when one feature is added to the subset, for each of the initial features. The feature whose addition most increased the classification performance is permanently added to the subset of selected features. Therefore, at the first iteration, the best subset with one feature is obtained. At the second iteration, the best subset of two features is obtained, under the constraint that one of the two features is the one selected in the first iteration, and so on. The algorithm is stopped once the accuracy is not significantly increased anymore by the addition of features, or when a chosen maximum number of selected features is reached.

In *backward elimination*, a similar iterative approach is considered, but we start instead from the full set of initial features, and then at each iteration the feature whose elimination is most favorable (or least unfavorable) is permanently eliminated from the selected features subset. The *bi-directional elimination* methods are based on a combination of these two wrapper algorithms: an initial subset of features is considered and at each iteration one feature is added through a forward selection iteration and one feature is eliminated by a backward elimination iteration.

Finally, *regularization* is another important approach to prevent overfitting due to high dimensionality and lack of training samples, and can also reduce the negative influence of noisy or redundant features. Instead of acting on the features, regularization acts on the training of a ML model in order to limit, and thus to tune, the complexity of the learnable models. The *regularization strength* is usually controlled by a parameter λ, which can be considered as a hyperparameter and optimized to maximize a classification performance estimation, such as the accuracy evaluated on validation sets in cross-validation. The most popular regularization technique, which is also the one used in the next chapters, is *L2 regularization*. The application of L2 regularization consists in adding the regularization term $\frac{\lambda}{2}||\mathbf{w}||^2$ to the cost function employed for training, where $||\mathbf{w}||^2 := \sum_j w_j^2$. E.g., the cost function in logistic regression with L2 regularization is (with reference to Eq. 1.8):

$$E(\mathbf{w}, w_0|\mathcal{X}) = -\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i) + \frac{\lambda}{2}||\mathbf{w}||^2$$

By adding a term proportional to the squared L2 norm of the weight vector (note that the intercept w_0 is not involved), we penalize the formation of weights with high absolute values during training. It should be noticed that this is indeed a way to reduce model complexity since, intuitively speaking, large weights allow to shrink the logistic function (Fig. 1.5) in the feature space, thus enabling sharper DB definitions, which in turn increases the model sensitivity to small details of the training sample population.

When L2 regularization is applied, it is good practice to apply *feature scaling* to the input samples, so that each feature takes on a similar range of values. Indeed, if some of the features had much smaller values than the others, these would require much higher weights so that they can properly influence the logistic regression output without being overshadowed by the larger features. In such a case, L2 regularization would therefore penalize the employment of the smaller features by limiting the magnitude of the learned weights. Therefore, to prevent this, feature scaling should be applied to the training samples before they are fed into the training algorithm. For example, *standardization* is a popular way of scaling the features, and it consists of subtracting from each feature value the corresponding mean and of dividing the result by the corresponding standard deviation. Both mean and standard deviation are estimated on the feature values comprised in the training samples. Moreover, also the feature values of the test samples should be scaled before they are employed for testing. However, the statistics required for this operation should be estimated using the training samples, to prevent learning from being influenced by information regarding the test samples.

1.6 Feature engineering and artificial neural networks

We have seen in Section 1.3 that calculating suitable features from the initial data might be essential to obtain a satisfying classification performance (e.g. see Fig. 1.9). Often, domain knowledge can be employed to do so, in what is called *feature engineering*. In our numerical example, we knew that the sample distribution had a sinusoidal shape with a specific angular frequency ω and therefore we engineered a suitable feature based on sinusoidal functions with the same angular frequency. Even in the case where we did not know the exact value of ω, we could still employ many sinusoidal features with different frequencies and, for instance, employ feature selection to find the most effective frequencies considering our training and validation sets. If a linear ML classifier such as logistic regression is used, the engineered features should create a feature space where the class-specific PDFs are linearly separable (or approximately so, depending on the target accuracy). In case this is not possible, we could still apply to the features a nonlinear classifier (e.g. consisting of the combination of several linear classifiers) that is able to automatically learn nonlinear DBs. In any case, feature engineering has the aim of creating a feature space where the subsequent classifier can better learn how to carry out the given task.

Let us consider an example related to the content of the following chapters: let us imagine that we want to classify different biological cell types from their microscope images, acquired while they are flowing in a microfluidic channel.

We know that, generally, a flowing cell has approximately a spherical shape, and that in the acquired images it is usually randomly displaced w.r.t. the center. Such a prior information can be useful to consider a relative small number of engineered features, and thus it provides a way to deal with the huge number of initial features given by the pixel values and to greatly simplify the classification task. In particular, we can consider the sample variability due to cell displacement and rotation as noise, since we know that it does not convey class-related information. This noise, which is likely to be stronger in magnitude than the useful information, can therefore be removed by employing displacement or rotational invariant (or approximately so) functions to extract features, such as the Fourier transform, auto-correlation or Hu moments (an example can be found in [3]).

However, in a ML problem we might not have sufficient domain knowledge to engineer effective features. The reason could be that we are not familiar enough with the problem, but in many cases the target task is so complex and the sample dimensionality so large that an accurate analysis of the given problem becomes exceedingly difficult. A way to overcome this issue is to employ a ML model that can automatically learn from the original data to extract suitable features. *Artificial neural networks* (ANNs) are very powerful and popular models, vaguely inspired by biological neural networks, that follow this principle of operation. *Multilayer perceptrons* are usually considered the most basic type of ANN. When applied to classification problems, intuitively speaking, they are built by applying linear classifiers to the outputs of several other linear classifiers, as shown in Fig. 1.16 (with reference to the linear classifier diagram shown in Fig. 1.7). Since all the operations from input to output are differentiable (excluding the final decision threshold), the derivative of the cost function w.r.t. each synaptic weight can be computed exploiting the chain rule, and thus all the weights can be trained by means of gradient descent (such a training method is called *backpropagation*). Operations corresponding to summing weighted inputs and applying a nonlinear function (generally called *activation function*) are called *output neurons* or *hidden neurons*, depending on whether their output is a final output of the whole network or not. Each hidden neuron can therefore learn a suitable representation of its input, tailored to improve the performance of the whole network. In this way, a suitable nonlinear DB can be efficiently drawn in the initial feature space without the need of feature engineering. Because of their complexity, ANN classifiers require a much larger number of samples to avoid overfitting, compared to linear classifiers. Moreover, while the cost function of a linear classifier (e.g. logistic regression) presents only one single global minimum given a training set, in multilayered perceptrons the cost function generally has several local minima. This makes the training of these models more difficult, since gradient descent is prone to converge to a local minimum, which might correspond to unsatisfactory

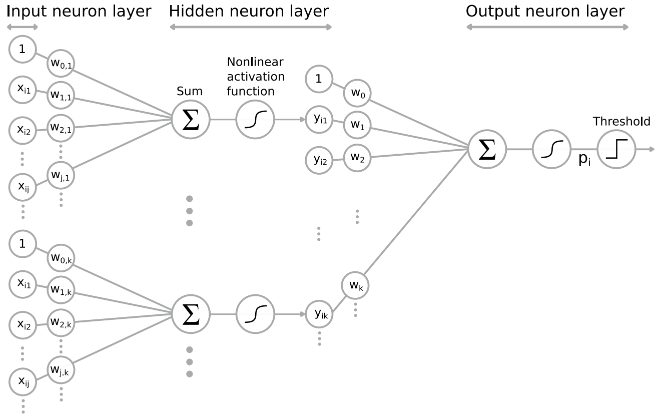


Figure 1.16: Architecture diagram showing the basic concept behind multilayer perceptrons (and ANNs in general): linear models are applied to the nonlinear function of the output of other linear models. In classification, linear classifiers (with reference to Fig. 1.7) are applied to the output of other linear classifiers. The ANN is divided in three main parts: the input neuron layer, which correspond to the initial input features; the hidden neuron layer, that learn suitable representation of the input data; and the output neuron layer. Each layer is connected to the subsequent one by synapses whose weights are determined by training. Deep learning architectures are characterized by several hidden neuron layers that can thus learn hierarchical representations.

performances.

If multiple hidden neuron layers are employed, hierarchical representations can be learned, allowing to carry out extremely complicated tasks such as object classification in images, sometimes even outperforming humans in accuracy. Such networks are called *deep* ANNs, and their training and use is referred to as *deep learning*. It should be stressed that in principle deep ANNs are not more powerful than ANNs with a single hidden neuron layer, if we do not put a constraint on the number of employed neurons. However, each neuron comes at a computational cost, and learning hierarchical representations usually allows deep learning to carry out complex tasks with a much lower number of neurons w.r.t. its shallow counterpart. This often comes with the price of many hyperparameters to tune (e.g. regarding the number of neurons in each layer, the number of layers or the structure of the connections), significantly complicating the training operation. Because of their extreme complexity, training deep ANNs from scratch might require a huge number of samples and might have a very high computational cost.

Prior information regarding the considered task can be incorporated into the ANN structure to significantly reduce the number of parameters. For example, considering the task of recognition of objects in images, the same type of object can appear in different positions and at different scales, depending on its position w.r.t. the camera. *Convolutional neural networks* (CNNs) are a special type of deep ANNs, which exploit these priors and are thus specifically structured for object detection in images [4].

1.7 The extreme learning machine approach

Another less popular ML approach that allows to avoid (or lessen the importance of) feature engineering, is to employ untrained random connections between the input neuron layer and a hidden neuron layer with a large number of neurons. Only the linear model at the readout (i.e. the weights of the connections between the hidden neuron layer and the output) is trained, thus eliminating the problem of local minima in the cost function and greatly reducing both the complexity and the computational cost of training. Moreover, backpropagation is not used and therefore the activation functions of the hidden neurons need to meet much less stringent requirements (e.g. they do not have to be differentiable). This enables the hidden neuron layer to create a large number of various random representations, which maps the input samples to a rich and high-dimensional feature space, to which the linear readout model (e.g. a linear regression for classification) is applied. An important requirement that remains for the activation functions, though, is that they must be nonlinear. We will further discuss this by means of a dedicated numerical example later on in this section.

This particular ML approach, which in the recent years was applied in various domains, is often referred to as an *extreme learning machine* (ELM) [5, 6], even though the main concept was proposed much earlier than when the name ELM first appeared, which raised related complaints [7]. Interestingly, the ELM approach is particularly suited for hardware implementations, where physical systems can be used to perform a large number of different and random nonlinear transformations of an input signal, without any computational cost. Indeed, in Chapters 3, 4 and 5 we consider the use of dielectric scattering media to project a random light intensity pattern from an impinging laser beam conveying the input information. We will see that by measuring such a pattern with an image sensor, we obtain a large number of pixel values that constitute a random nonlinear mapping of the input information. Then, a linear classifier (logistic regression) is applied to the pixel values and trained to carry out a classification task. It should be stressed that since nonlinear functions are usually significantly more computationally expensive than linear ones, using a fast physical system to perform the random activation function operations in the hidden layer can greatly enhance inference speed and throughput.

Let us now apply the ELM concept to our numerical example, by employing a relatively large number of different nonlinear features, without exploiting any prior information for feature engineering. In particular, we create new features raising the original ones $x_{i,1}$ and $x_{i,2}$ to the power of few natural numbers: $x_{i,1}^n$ and $x_{i,2}^n$, with $n = 1, 2, 3, 4, 5, 6$. Moreover, we also consider all the possible combinations given by their multiplication: $x_{i,1}^n x_{i,2}^m$, with $m = 1, 2, 3, 4, 5, 6$. Thus, in total we create 48 new features through different nonlinear operations on the original ones. Since we employ a large number of features, we expect to easily incur overfitting. Indeed, using 50 training samples per class, we obtain a flawed DB (Fig. 1.17 *a*, *b*). However, increasing the number of samples to 200, we obtain quite an accurate DB (Fig. 1.17 *c*). Comparing the corresponding learning curve (Fig. 1.18) with the one obtained adding sinusoidal features (Fig. 1.15 *a*), we notice that similar results are obtained (some discontinuity in the learning curve of Fig. 1.18 is due to failure of the employed solver to converge). An important difference between the two cases, is that among the sinusoidal

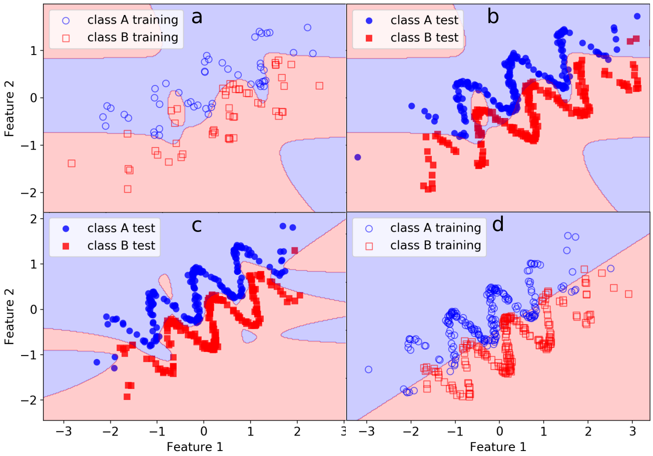


Figure 1.17: Classification with logistic regression. *a*: 48 different nonlinear features were created through exponentiation and multiplication. Due to overfitting (50 training samples per class are used), the corresponding learned DB shows reduced generalization when applied to test samples (*b*). *c*: substantially better generalization is obtained using 200 training samples per class. *d*: if 48 features are created through linear operations, no advantage is obtained over the linear discrimination learned considering only the initial features.

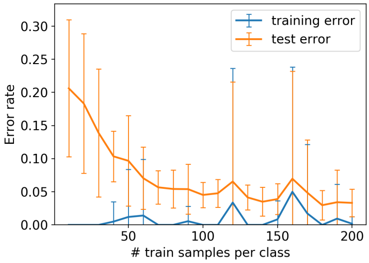


Figure 1.18: Learning curve considering 48 different nonlinear features created through exponentiation and multiplication, corresponding to the plots in Fig. 1.17 *a,b,c*.

features, that can be considered as engineered since the choice of their form was based on prior information, there were two features (the ones with $\tilde{\omega} = \omega$) that alone could already generate a good DB. Instead, none of the features created through exponentiation could provide, alone or in pairs, such a good separation. Nevertheless, a linear combination of several of them could define a good DB, similarly as a sinusoidal function can be locally approximated by its truncated Taylor series expansion.

Until now we have always created new features through nonlinear operations and we have stated that in ANNs, the activation function of the neurons must be nonlinear. Indeed, this nonlinearity is what defines the separation between different neuron layers and therefore it is key to generate hierarchical input representations in deep learning. An explanation for the importance of nonlinearity in feature extraction or expansion, is that if a feature is obtained by means of a linear operation on other initial features, the sample vectors will become linearly dependent even if originally they were not. Therefore, even though formally the dimension of the considered feature space increases by one, the new vectors will still lay on a hyperplane with the same initial dimensionality and their relative reciprocal positions remain unchanged. Thus, the problem of finding a linear DB that separates the classes is practically the same as in the previous feature space. In simpler words, the newly added feature conveys redundant information, from the perspective of a linear separation problem. Intuitively, since a linear classifier learns the optimal linear transformation of its input features, prior linear feature creation cannot do something more than what the ML model can already learn to do.

To show this concretely with our numerical example, let us create the same number of features, by performing the following linear operations on the initial features: $nx_{i,1}$, $mx_{i,2}$ and $nx_{i,1} + mx_{i,2}$, with $n = 1, 2, 3, 4, 5, 6$ and

$m = 1, 2, 3, 4, 5, 6$. As expected, we obtain a linear DB in the initial feature space (Fig. 1.17 *d*), indicating that the linear feature expansion has not provided any advantage.

1.8 Summary and conclusion

In this chapter we first covered some basic definitions and aspects regarding machine learning classification and we introduced a mockup problem that we used to exemplify the main concepts encountered across the whole chapter. Then we presented the logistic regression model, which we employ throughout all the following chapters, and its application to linear classification. Afterwards, we discussed the problem of overfitting, that is almost ever-present in real-life machine learning implementations, and the important practice of validating the performance of a trained model. We saw how overfitting can be monitored by drawing a learning curve and how the training can be optimized through hyperparameter tuning with a limited number of available samples. Then we briefly presented a few common techniques to control overfitting by means of dimensionality reduction and regularization. Furthermore, we discussed classification performance improvement through feature engineering, and how this is automatically learned by artificial neural network models. Finally, we briefly introduced the extreme learning machine approach, which is based on the creation of a large number of random nonlinear features to achieve high classification performance with a very simple training process. The extreme learning machine paradigm is particularly suitable for efficient hardware implementations, where a physical system is employed to perform the required nonlinear transformations of the input. For this reason, we consider this approach in Chapters 3, 4 and 5, to perform biological cell and particle classification in imaging label-free flow cytometry at a very low computational cost.

References

[1] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2014.

[2] Sebastian Raschka. *Python Machine Learning*. Packt Publishing, 2015.

[3] Yuqian Li, Bruno Cornelis, Alexandra Dusa, Geert Vanmeerbeeck, Dries Vercruysse, Erik Sohn, Kamil Blaszkiewicz, Dimiter Prodanov, Peter Schelkens, and Liesbet Lagae. *Accurate label-free 3-part leukocyte recognition with single cell lens-free imaging flow cytometry*. Computers in biology and medicine, 96:147–156, 2018.

[4] Waseem Rawat and Zenghui Wang. *Deep convolutional neural networks for image classification: A comprehensive review*. Neural computation, 29(9):2352–2449, 2017.

[5] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. *Trends in extreme learning machines: A review*. Neural Networks, 61:32–48, 2015.

[6] Weipeng Cao, Xizhao Wang, Zhong Ming, and Jinzhu Gao. *A review on neural networks with random weights*. Neurocomputing, 275:278–287, 2018.

[7] L. P. Wang and C. R. Wan. *Comments on "The Extreme Learning Machine*. IEEE Transactions on Neural Networks, 19(8):1494–1495, 2008.

2

A machine learning study on efficient white blood cell classification in imaging label-free microflow cytometry

In this chapter we describe our work in collaboration with our project partners from the Interuniversity Microelectronics Centre (IMEC), who developed a prototype of an *on-chip single-cell flow cytometer* for *label-free white blood cell sorting*, based on *lens-free digital holographic microscopy* [1–3] (in Chapter 0 an introduction on these topics is provided). Employing such a setup, which is described in the next section, our partners generated thousands of cell interference patterns (also called *cell holograms*) from three different types of white blood cells (WBCs). Our goal was to demonstrate a computationally efficient software implementation to classify the cell holograms on the basis of the cell type. However, we discovered that the influence of drift in measurement conditions could bias the learning process of the considered machine learning classifiers, preventing the development of a generalizable solution. We then investigated this issue in detail, by means of several machine learning experiments.

2.1 Experimental setup

Since a more detailed description of the setup is provided in [1–3], we here only describe the main components and functionalities. Three different WBC types (monocytes, T cells and granulocytes) are made to flow in a microfluidic channel (Fig. 2.1). Laser light ($\lambda = 532\,\text{nm}$) is shone through an aperture perpendicularly to the channel and the resulting diffraction pattern is projected on an image sensor. The presence of a flow cell, whose light absorption is negligible, modifies the interference pattern acquired by the sensor. The obtained image, called cell hologram, carries information about the 3D refractive index structure of the cell, which can be used to reconstruct the cell image (this is called digital holographic microscopy). This operation is computationally expensive, thus in this work we circumvent this problem by skipping image reconstruction. Rather, we directly employ raw cell holograms for the machine learning classification of the WBC type. Indeed, the classification algorithm needs to run fast enough so that a result can be provided to the cell sorter before the imaged cell reaches it after passing through the channel. Therefore, the computational cost of cell classification poses a direct limit to the throughput of this type of cell sorters, especially if a large number of those are to be integrated on a single chip.

The imaging system is designed to capture the optical path distortion due to a fast flowing microscopic transparent object such as a cell, without the use of lenses, as opposed to traditional microscopes. The holograms produced by this device can be extremely sensitive to small changes in measurement conditions over time, e.g. due to fluctuations of the light source properties, displacement or distortion of the optical beam (e.g. due to thermal expansion of some elements), refractive index changes of the optical components (e.g. due to slow water absorption of the microfluidic channel walls) and so on. A way to mitigate the detrimental effects of these processes is to subtract from each acquired image containing cell information a corresponding background image (i.e. without cell information) acquired right before or right after. As we will see later on in this chapter, such a *background subtraction* technique alone might not be enough to ensure that drifting measurement conditions do not undermine the training of a machine learning classifier.

In order to limit the number of measured holograms, and in order to make sure that each cell has a corresponding temporarily close background image, the flow cytometer is also equipped with a fluorescent detection stage positioned along the channel before the described imaging stage. The fluorescent detection stage is structurally similar to the imaging stage: laser light ($\lambda = 488\,\mu\text{m}$ for fluorescent excitation) is shone through a pinhole into the channel, and after passing through an optical filter that selects the fluorescence wavelength, it reaches a photomultiplier tube. This stage is employed to detect the passage of

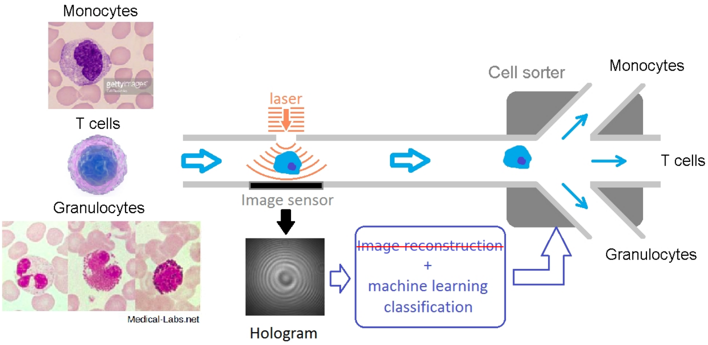


Figure 2.1: The main functionalities of the flow cytometer (developed by IMEC) that generated the WBC holograms considered in this work. Three different WBC types (monocytes, T cells and granulocytes) are made to flow in a microfluidic channel. Laser light ($\lambda = 532\,\mu m$) is shone through an aperture perpendicularly to the channel and the resulting diffraction pattern is projected on an image sensor. The presence of a flowing cell, whose light absorption is negligible, modifies the interference pattern acquired by the sensor. The obtained image, called cell hologram, carries information about the 3D refractive index structure of the cell, which can be used to reconstruct the cell image (digital holographic microscopy). This operation is computationally expensive, thus in this work we directly employ raw cell holograms for the machine learning classification of the WBC type.

the cell type that was previously marked with a selective fluorescent stain. It can trigger the imaging stage so that only the target cells are imaged, together with a corresponding (temporally close) background image. It should be stressed that this fluorescent detection stage is not employed to classify the cells, but only to detect the single cell type that was marked.

The cell sorter is built on the same microfluidic chip, and relies on micro-heaters to quickly generate vapour bubbles and thus create liquid jets that push a cell into one of the three channel outputs (one per cell type), depending on the obtained classification outcome. It should be stressed that even if the cells are stained with a fluorescent label, this is only used to provide examples for the learning of the machine learning classifier, which would operate in a label-free way once trained. In particular, the employed cells are taken from fresh blood samples, from which the red blood cells were removed. In each blood sample, a different type of WBC (namely lymphocytes, monocytes and granulocytes) was stained with fluorescent label, so that only images of cells of a single type are acquired when a blood sample flows in the cytometer. This step was necessary to provide the ground-truth (labels) required for the classifier training.

2.2 Preliminary analysis of the background illumination properties

Our partners from IMEC provided us with a total of 57,064 images of 2048×1088 pixels, corresponding to a data volume of around 120 GB. The images are divided in three groups, obtained from separate measurements, one for each cell type: 20,797 images from the monocyte measurement, 3,753 images from the T cell measurement and 32,514 images from the granulocyte measurement. As it was explained in the previous section, each group comprises both cell holograms and background images. Because the cells are flowing together in the microfluidic channel when their hologram is acquired, the holograms of the same cell type can undergo several variations, mainly regarding the cell position but also, for example, due to its interaction with other cells or due to reflections (Fig. 2.2). Moreover, we noticed that the holograms of the three cell types are not easily distinguishable by human inspection (e.g. see Fig. 2.3), also considering that the cell hologram appearance is strongly dependent on the background illumination, which seems not to be constant.

Before choosing a machine learning strategy, it is important to investigate some general properties of the available data, e.g. the presence of different kinds of noise or how class-related information is represented. Since we deal with high-resolution images, a thorough analysis employing the raw original data requires long computation times and could overload the computer memory. Therefore, we chose to transform the 2D matrices (i.e. the images) into 1D arrays, obtained by summing all the matrix elements (i.e. the pixel values) in each column. In other words, the images are integrated along the cell flow direction obtaining what we call an *1D hologram* (Fig. 2.4). Note that the sample dimensionality and its data storage are reduced by around three orders of magnitude. This choice is justified by the fact that both the background illumination and the cell pattern approximately present a cylindrical symmetry, therefore the most important properties are still easily deducible from their 1D integration. A similar approach was indeed employed in a previous work on an analog dataset generated by simulations [4], where a pixel row at the center of the cell pattern was considered. Moreover, we cropped the images to exclude the dark parts outside the illuminated area and we subtracted the constant average noise level present in these regions.

The background illumination draws a bell-like curve in the 1D hologram images (Fig. 2.4), as expected from a Gaussian beam. However, the background shows visible variations from one measurement to another (Fig. 2.5 *a*), but also continuous drift in intensity during the same measurement (Fig. 2.5 *b*). It should be stressed that trivial variations in illumination intensity could be easily compensated for by normalizing the acquired holograms, while other kind of varia-

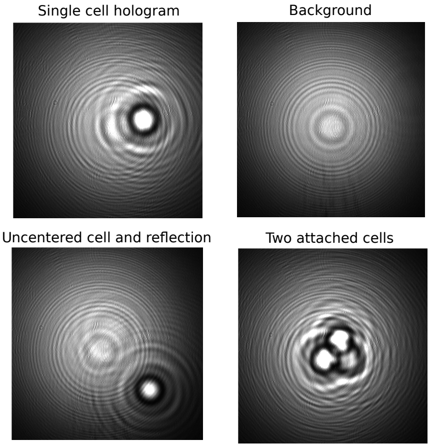


Figure 2.2: Examples of hologram images obtained from the described cytometer. A cell hologram is mainly given by the overlapping of the light diffraction originated at the pinhole (the aperture from which the light is shone into the channel, which provides the background pattern) and by the cell. The examples show how the acquired images may vary depending on different scenarios. The illuminated cell can be more or less centered w.r.t. the background pattern and additional faint patterns might appear due to reflections on the channel walls. Moreover, two or more attached cells can generate very different patterns that may complicate the classifier training.

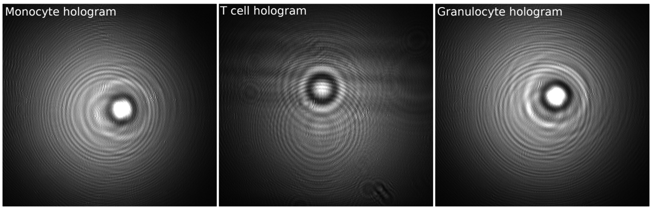


Figure 2.3: Examples of holograms of the three cell types obtained from the described cytometer.

tions such as in background position or shape can be more difficult to treat.

In order to shed light on the type and magnitude of the background variations, we need to automatically extract the relevant information from a high number of 1D holograms. A solution is to fit each of them to an analytical func-

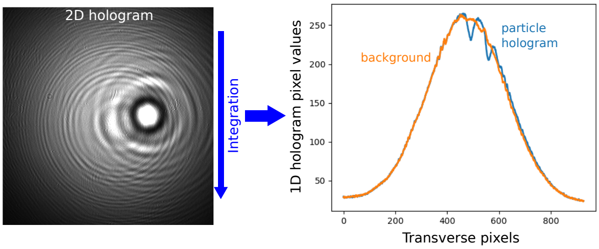


Figure 2.4: By summing the pixels values along the channel direction, a 2D hologram image (*left*) is transformed into a 1D hologram (*right*). The plot at the right shows a comparison between two example 1D holograms: the particle information is represented by a perturbation of the background bell-like profile.

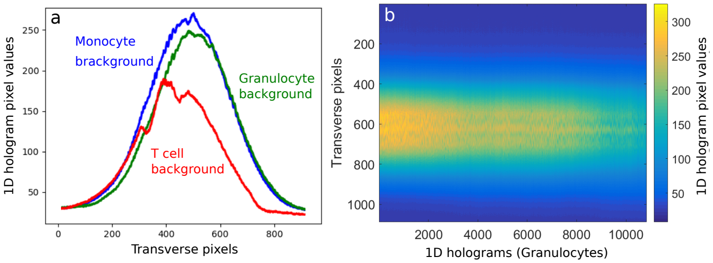


Figure 2.5: *a*: three examples of background 1D holograms, each from a different measurement corresponding to the cell class: monocytes, granulocytes and cells. It is evident that the background illumination undergoes significant variations in intensity, shape and center position. *b*: color map of the 1D holograms acquired in a part of the granulocytes measurement. The 1D holograms are stacked horizontally and chronologically ordered along the x-axis. It can be noticed that the overall intensity gradually decreases over time.

tion, from which it is possible to deduct quantities such as the illumination center and the width of the bell-like shape. To this end, we find that a Gaussian function $f(x) = a\exp(-(x-b)^2/(2c^2))$ can provide a good fit for the background shape, where a accounts for the overall intensity of the background illumination, b for its center of and c for its width. To check how good the fit is for each hologram, we calculated the relative quadratic error, i.e. the sum of the squared point-wise difference between the fit and the actual hologram, divided by the hologram overall intensity (Fig. 2.6 *a*). It can be noticed that the average distance of the 1D hologram to a fitted Gaussian function significantly depends on the class-specific measurement. In particular, the holograms from the T cell measurements seem to differ from a Gaussian to a higher degree w.r.t. the holograms from the granulocyte measurement, which in turn shows a higher average fitting error w.r.t. the holograms form the monocyte measurement. Evidently, this originates from changes in shape of the background illumination, since both cell images and background images were fitted.

Generally, we could visually check that the Gaussian function still provides quite a good fit for the background illumination shape (e.g. in Fig. 2.6 *b*). Indeed, even for the fits with the highest errors, the discrepancy is mainly given by the perturbation caused by the cell presence, and the Gaussian function still seems to follow the underlying background shape (e.g. in Fig. 2.6 *c*). From these observations we can conclude that the performed Gaussian fits provide a good approximation of the background illumination, even when the hologram is perturbed by the cell presence.

Analysing the calculated parameters of the fitted Gaussian function for each hologram, we obtained a clearer view of how the background intensity and displacement (w.r.t. the image center) changed over time within each measurement session (Fig. 2.7 *a* and *b* respectively). In particular, it can be noticed that these parameters drifted very significantly during the measurement sessions. Notably, the slow drifts are much stronger than the fast and noisy fluctuations due to the cell pattern presence and variability. Moreover, also the background displacement drifts significantly, spanning a range larger than 40% of the background width at half maximum (Fig. 2.7 *b*). These observations suggest that it might be problematic to use these samples for the training of a machine learning algorithm that classifies the cell types, since it might be difficult for the classifier to distinguish between the strong data variations due to the background and the variations due to the cell types differences.

Unfortunately, because of technical reasons, our collaborators could not investigate the causes of the permanent differences in background shape nor the drifts in illumination properties. However, our own flow cytometry experiment (presented in Chapters 4 and 5) showed later on that it is difficult to avoid significant drifts in the acquired holograms, because of their high sensitivity to measurement conditions.

2.3 Labeling of background images

As explained in the setup description, in each measurement session the experiment was designed to acquire the holograms of a specific type of cell together with the neighbouring background holograms (without cell information). However, in the data set it is not uncommon that images that are supposed to contain

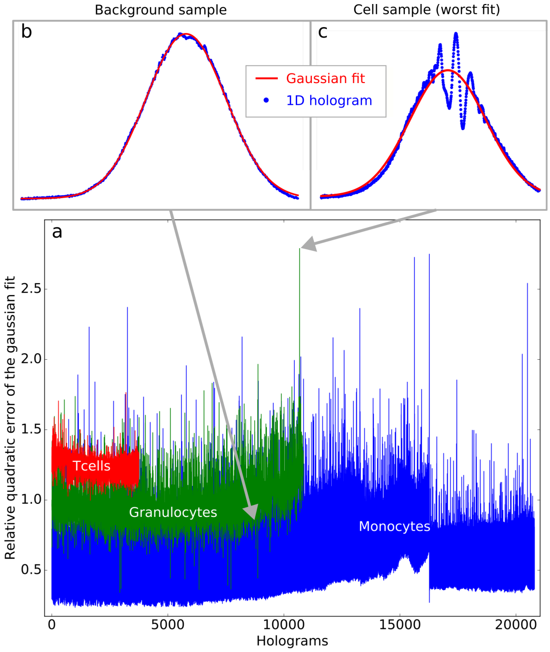


Figure 2.6: The available 1D holograms from the three measurements sessions (monocytes, granulocytes and T cells) were all fitted with a Gaussian function. *a*: fit errors (y-axis) calculated as the sum of the squared point-wise difference between the fit and the actual hologram, divided by the hologram overall intensity. The x-axis provides the index of the acquired holograms, in chronological order w.r.t. to the corresponding measurement session. *b* and *c*: Gaussian fit examples respectively of a background 1D hologram from the granulocyte measurement (see light grey arrow) and of the hologram with the highest fit error (due to the presence of a cell). Generally, the Gaussian fit provides a good approximation of the background illumination. However, it can be noticed that goodness of fit significantly depends on the measurement session.

only the background show instead a cell hologram. Since we planned to consider the background images as a separate class in the classification, we had to find a way to accurately detect background images in order to obtain a suitable ground truth to train the machine learning classifier.

In order to automatically and accurately label the background holograms, we needed to find a suitable function whose outcome is very sensitive to the

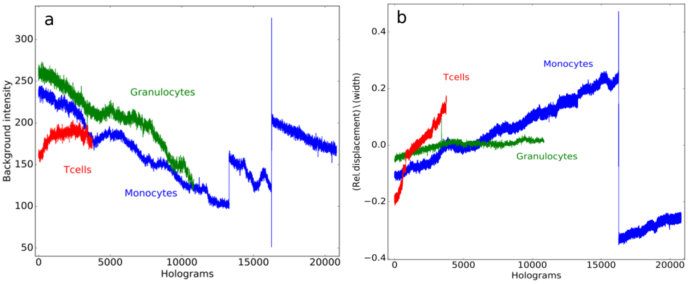


Figure 2.7: Background illumination properties extracted from the Gaussian fit of the available 1D holograms. The x-axis provides the index of the acquired holograms, in chronological order w.r.t. to the three corresponding measurement session. *a*: the background illumination intensity drifts slowly and significantly over time during each measurement. *b*: background displacement divided by the background width at half maximum. Generally, the background illumination undergoes slow and large drifts in intensity and position during the measurements.

perturbations due to the cell presence and less sensitive to the strong variations in background illumination previously presented. For example, as suggested by Fig. 2.7 *a*, the overall hologram intensity would not be an appropriate function to this end. After trying few options, such as the goodness of fit shown in Fig. 2.6 *a* and the Fourier transform, we found that a good separation between cell and background samples was provided by the normalized *auto-correlation function*, here defined as the Pearson correlation between a 1D hologram h_x and its copy translated by Δx along the x axis:

$$A_h(\Delta x) = \frac{\langle h_x h_{x+\Delta x} \rangle - \langle h_x \rangle^2}{\langle h_x^2 \rangle - \langle h_x \rangle^2} \tag{2.1}$$

Here $\langle f_x \rangle$ represents the arithmetic mean of the discrete function f_x over the x values. Since the holograms are discrete functions of the pixel index x, the translation Δx is discrete and takes on a range of natural numbers $0, 1, 2, ..., N_x - 1$, where N_x is the number of pixels of the 1D holograms. To make sure that the arrays h_x and $h_{x+\Delta x}$ have the same number of values, they are truncated so that the last Δx pixels are left out in h_x and the first Δx pixels are left out in $h_{x+\Delta x}$. As we will see later on, in this work we will consider small translations w.r.t. the total number of pixels in the 1D holograms, i.e. $\Delta x \ll N_x$. This guarantees that a negligible information loss is caused by the truncation operation, considering that the borders of the employed holograms correspond to noisy and irrelevant regions where the illumination is low.

To get some insight into why the auto-correlation can be useful in detecting the cell perturbations in the holograms, we can notice that an undulated perturbation such as the one caused by the cell presence (see Fig. 2.4) causes a decrease in the auto-correlation values for $\Delta x \approx T_p/2$, where T_p is, loosely speaking, the period of the undulated perturbation. At the same time, the auto-correlation of a 1D hologram is negligibly affected by displacement or change in width of the background illumination, given that Δx is small enough so that only the irrelevant weakly illuminated borders of the image are truncated. Moreover, the normalization in Eq. 2.1 ensures that the auto-correlation range spans from -1 (completely anticorrelated) to 1 (completely correlated), removing the influence of the background overall intensity. In summary, the auto-correlation is sensitive to cell presence and robust against the background illumination drifts shown in Fig. 2.5.

Indeed, when plotting the sum of the first 24 auto-correlation values (i.e. corresponding to Δx = 1, 2, 3, ..., 24) for each hologram it becomes clear that these values are clustered in two groups: one with higher auto-correlation values and lower variance that corresponds to the background images (without cell presence) and another with lower values but larger variance corresponding to the cell holograms (Fig. 2.8). The background-cell separation was visually validated by manually checking several samples throughout the whole data set. The number of the summed auto-correlation values (i.e. 24) was chosen to optimize the separation after trying several options.

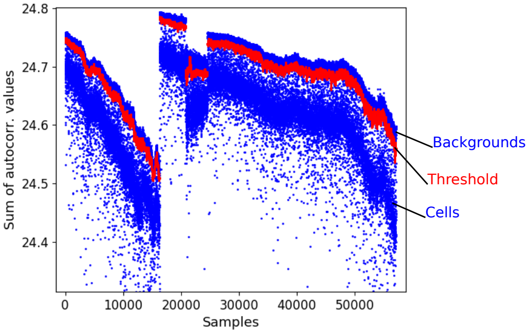


Figure 2.8: Sum of the first 24 auto-correlation values (blue dots) as a function of the employed 1D hologram indices. This function separates background and cell holograms in two disjointed clusters (upper and lower respectively). The red dots represent the moving threshold values used to distinguish background samples from cell samples.

We implemented an automatic background labelling algorithm that is based on a moving threshold (red dots in Fig. 2.8) calculated as a suitably calibrated linear combination of a moving average and a moving variance of the auto-correlation summed values. An effort was made to make the threshold algorithm robust against outliers by removing the influence of points lying outside the range centered on the moving average and with a width of four times the moving standard deviation. Such an automatic labeling method has proven to be effective and reasonably accurate throughout the whole work on the cell holograms. From the available images we obtained 10896, 1991, 13132 and 31124 holograms respectively of monocytes, T cells, granulocytes and background.

Once we were able to separate backgrounds from cell holograms, we could plot the average background illumination for the measurements corresponding to the three different types of cell (Fig. 2.9). It can be noticed that the illumination pattern in the T cell measurement significantly differs in shape w.r.t. the others.

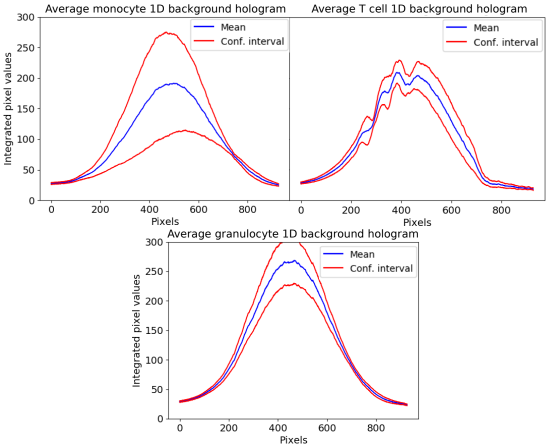


Figure 2.9: Average background illumination for the measurements corresponding to the three different types of cell. The confidence interval represented by the red lines corresponds to twice the point-wise standard deviation of the 1D holograms. It can be noticed that the illumination pattern in the T cell measurement significantly differs in shape w.r.t. the others.

2.4 Cell classification and the problem of measurement bias

2.4.1 Machine learning approach: feature extraction and pipeline

In this section we present a first stage of our work regarding the machine learning classification of the described cell holograms. We chose to consider 1D holograms as a starting sample set on which we developed our machine learning approach. In addition to the motivations previously given in Section 2.2, we should point out that this data representation can simplify the considered cell classification task by removing the variability due to the cell hologram displacement along the channel direction. In fact, sample variability that is not directly generated by class-specific characteristics is to be considered as noise from the perspective of machine learning classification. Depending on its magnitude and properties, noise can significantly disrupt the learning process to the point that the target classification performance is not achievable or that a more powerful or better suited classifier model is required. In our specific case, as we discussed in Chapter 0, we aim to develop an effective classifier that requires as low computational power as possible, in order to allow for high-throughput operations. Therefore, we pay particular attention to dealing with noise as efficiently as possible, possibly discarding powerful but heavy models such as convolutional and/or deep NNs.

It is also interesting to point out that the transformation of the originally acquired 2D images into 1D holograms can reduce the required computational resources not only because of the much lower number of pixels, but also because it can help lowering the cell-displacement noise affecting the samples. Moreover, 1D holograms are similar to the 1D image acquired by a line-scan sensor (i.e. like an image sensor, but with only one row of pixels, see Fig. 2.10) when used instead of a camera, which could significantly speed up the online cell classification due to the much higher frame rate. Also for this reason it would be interesting to demonstrate that fast and accurate cell classification could be performed considering 1D holograms.

If variability due to hologram displacements along the channel direction is suppressed by passing from a 2D to a 1D representation, the hologram displacement along the transverse direction still introduces a significant amount of noise. This is true for both the cell displacement (e.g. see Fig. 2.2) and the background illumination displacement (see Fig. 2.7 *b*). As we discussed in Section 2.3, the auto-correlation function (Eq. 2.1) is robust to (i.e. can suppress) both displacement and scale variability, and therefore we considered its values as promising feature candidates. For the same reasons, we also considered the application of Fast Fourier Transform (FFT) to the hologram, calculated through the `numpy.fft.rfft()` function from the Numpy Python library. Our approach, which is common in machine learning, is to extract useful features from the hologram samples and to feed these to a linear classifier (see Chapter 1 for details) that learns the best linear combinations of the provided features so that the outcome accurately distinguish samples belonging to different classes (see Fig. 2.10). In particular, we employ logistic regression (explained in Chapter 1) as a linear classifier model.

To give an example, the absolute value of the FFT applied to the hologram array (i.e. the hologram spectrum) provides information about how a cell influences the frequency components of the background hologram. If cells of different types systematically introduce class-specific changes in some of the obtained FFT values, the linear classifier is expected to learn to combine these values so that its outcome indicates which class the analyzed cell belongs to.

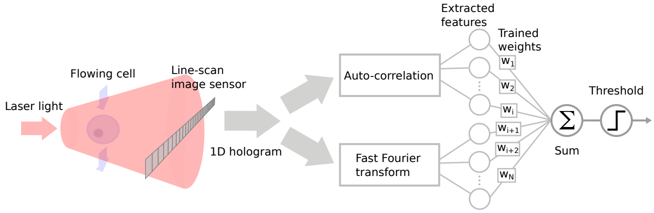


Figure 2.10: The proposed classification pipeline. We obtain 1D holograms by summing 2D hologram pixels along the flow direction, which is an approximation of the use of a line-scan image sensor. Suitable auto-correlation and FFT values are calculated from the 1D hologram (feature extraction). These features are then weighted and summed by a linear classifier (logistic regression). An optimal set of weights for each class is learned during training. The weighted sum with highest outcome indicates the cell class recognized by the classifier.

Let us now outline some common aspects of the employed machine learning pipelines treated in this chapter, which will be taken for granted from now on. We *labeled* the holograms as belonging to six disjointed classes, depending on which of the three measurements a hologram was generated from and on whether it was labelled as background or not:

- *Monocyte cell.*
- *T cell.*
- *Granulocyte cell.*
- *Monocyte background.*

- *T cell background.*
- *Granulocyte background.*

We will be mainly considering two types of classification:

- *Cell classification*, where the 4 considered classes are monocyte, T cell, granulocyte and background; the last one comprises all the samples labelled as background and is therefore the union of the three background classes.
- *Background classification*, where the 3 considered classes are monocyte background, T cell background and granulocyte background. This will be relevant when evaluating the bias present in the data.

The considered sample sets were then shuffled and combined to form a *training set* (around 75% of the samples) and a *test set*, respectively used for the training and for the final evaluation of the classifier. While doing this, we made sure that each set contained more or less an equal percentage of differently labeled samples. Following the machine learning good practice to avoid undetected overfitting (see Chapter 1), we paid attention never to employ any information regarding the test set in order to make any choice on the classification model under evaluation. As a linear classifier, we employed the `linear_model.LogisticRegression()` classifier provided by the *Scikit-learn* Python library for machine learning [5]. The optimization of hyperparameters, such as regularization strength or feature selection (see Chapter 1 for machine learning definitions and explanations), was performed through 6-fold cross validation on the training set. Confusion matrices were calculated on the test set to provide a final performance evaluation of the employed classification pipeline. Moreover, every time a significantly different pipeline was considered for evaluation, the learning curve was drawn to check whether the number of available training samples was suitable.

2.4.2 First classification attempt reveals presence of measurement bias

In the previous section, we have seen that the sum of the first 24 auto-correlation value of the 1D holograms could successfully separate background samples from cell samples. Our first classification attempt consisted in employing these 24 auto-correlation values as features (see an example of these features in Fig. 2.11). We decided to consider class-balanced sample sets, randomly selecting 1191 samples (i.e. the number of available T cell holograms) from each class, considering all the background images as one class. In this case we chose not to use all the available samples because the learning curve showed that it was not necessary,

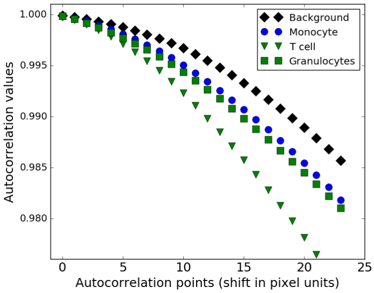


Figure 2.11: Example of auto-correlation values for small pixel shifts. Each marker type corresponds to a randomly selected sample from the set indicated in the legend. In Section 2.3 the sum of these values was used to distinguish between background and cell holograms. In this section, these values are employed as features for the machine learning classifier.

which was indeed expected given the small number of employed features. After the classifier was trained, we tested it on the test sample set, obtaining a satisfying test accuracy (see Fig. 2.12 a).

However, in Section 2.2 we saw that the background illumination undergoes significant changes during each measurement session and between one session and another. This makes us wonder how these changes influence the classification performance and, in particular, if our machine learning algorithm learns the classification task by exploiting the differences introduced by the measurement conditions rather than on the basis of the distinguishing traits of the different cell types. Indeed, if this was the case, we could not trust our classifier to perform well in a practical application, i.e. when employed to distinguish the different cell types in a single measurement session.

The best way to check this possibility would be to perform new measurements to provide additional test samples, so that the measurement conditions of train and test sets would be uncorrelated. However, because of technical reasons, it was difficult for our project partner to provide us with new samples. Therefore we tried another approach, that is to train our algorithm to classify background images from different measurement sessions and without any cell information, i.e. considering the *monocyte background*, *T cell background* and *granulocyte background* classes. Intuitively, if the classifier were completely unable to learn to classify the backgrounds, it would be unlikely that the classifier could exploit measurement condition information to illicitly improve its performance evaluation on the cell classification task. Unfortunately, we found that

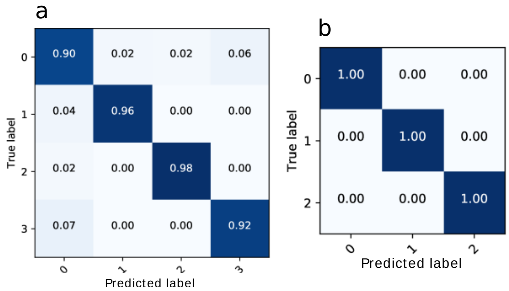


Figure 2.12: Confusion matrices showing test accuracy fractions for each class. The first 24 auto-correlation values of 1D holograms were employed as features. *a*: Full cell classification. Labels 0, 1, 2 and 3 respectively stand for *background*, *monocyte cell*, *T cell cell* and *granulocyte cell* classes. High classification accuracy is achieved (see diagonal). *b*: Background classification based on which measurement session they were generated from. Labels 0, 1 and 2 respectively stand for *monocyte background*, *T cell background* and *granulocyte background* classes. The classification shows no test error, implying that the measurement conditions influences the background holograms so that the classifier easily detects from which measurement session they are from, i.e. *measurement bias* is revealed.

our classifier could learn to classify backgrounds without error (see Fig. 2.12 b), suggesting that the classification performance evaluation shown in Fig. 2.12 a is significantly biased, so that a misleadingly high accuracy is obtained.

In this work, this particular type of overfitting is referred to as *measurement bias*, defined as the bias in the training of a machine learning model due to the correlation between the noise in the training samples and the corresponding labels. Generally, measurement bias decreases the evaluated classification performance if such a correlation is broken in the test samples, and it artificially boosts test accuracy in an elusive and non-generalizable way. It should be stressed that this is a well-known problem in machine learning, which falls in the broader category of *dataset bias* or *shortcut learning* [6]. Therefore, taking suitable precautions to prevent it is considered good practice. However, in reality many papers about cell classification with machine learning do not report any specific treatment for this problem. In Chapter 5 we study measurement bias in an ad-hoc microflow cytometry experiment and we demonstrate a practical solution.

2.5 Attempts at bias-free classification

2.5.1 Background subtraction

A straightforward way to try to remove the influence of the measurement conditions from the samples, i.e. to solve the measurement bias problem discussed in the previous section, is background subtraction. Generally, in imaging flow cytometry, background samples are easily generated and it is convenient to work on images obtained by the difference between cell images and background images, e.g. as in [2, 3]. In order to try to remove the influence of drifting measurement conditions, backgrounds are acquired as temporally close as possible to the cell samples from which they are subtracted. We performed this operation by removing from each image the chronologically closes image labelled as background (see examples of background-subtracted 1D holograms in Fig. 2.13 a). It should be stressed that when computationally efficient classification is targeted, the requirement of software-based background subtraction might introduce a significant disadvantage, since it implies the additional operations of storing and subtracting a previously acquired image from each image.

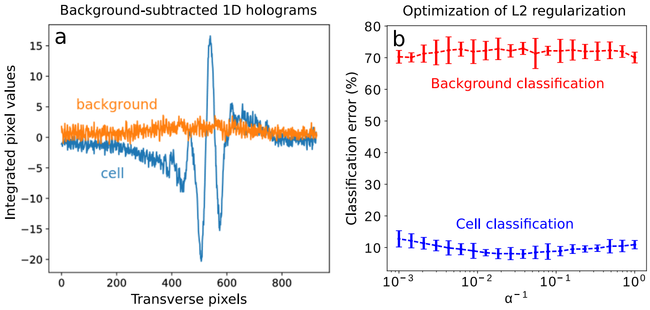


Figure 2.13: *a*: Example of background-subtracted 1D hologram of a cell image and of a background image. *b*: Classification error (evaluated through cross validation) as a function of the inverse of the L2 regularization strength. In this case, the linear classifier was applied directly to the pixel values of background-subtracted 1D samples. The error bars correspond to twice the standard deviation of the error estimations. Good accuracy was obtained in cell classification (blue), while the high error of background classification (red) suggests that biasing information was removed from the background samples.

In order to check if background subtraction could remove measurement bias, we first tried a “naive” classification approach: we skipped feature selection and we applied the linear classifier directly on the pixel values of background-subtracted images. Since we employed much more features, and presumably more noisy ones, w.r.t. the previously considered 24 auto-correlation values, we employed the L2 regularization (see Chapter 1 for details) to reduce overfitting. The regularization strength α was considered as a hyperparameter and was optimized through cross-validation. We obtained a surprisingly high overall accuracy in cell classification (error < 10%, see Fig. 2.13 b). Moreover, the background classification shows an error higher than 70% (the expected random guess error for three classes is $66.\overline{6}\%$), suggesting that biasing information was removed, at least from the background images. Employing the optimal L2 regularization strength, i.e. the one that minimizes the validation error for cell classification, we obtain promising confusion matrices evaluated on the test samples (see Fig. 2.14).

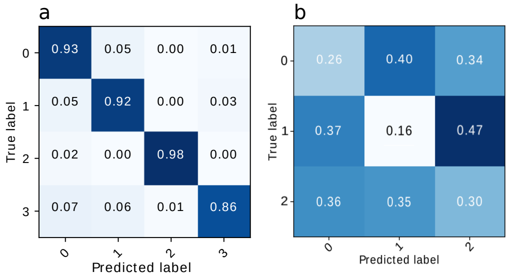


Figure 2.14: Confusion matrices showing test accuracy fractions for each class. The pixel values of background-subtracted 1D holograms were employed as features. *a*: Full cell classification. Labels 0, 1, 2 and 3 respectively stand for *background*, *monocyte cell*, *T cell cell* and *granulocyte cell* classes. *b*: Background classification based on which measurement session they were generated from. Labels 0, 1 and 2 respectively stand for *monocyte background*, *T cell background* and *granulocyte background* classes. The classification shows that the test error is homogeneously distributed, suggesting that background bias was removed from the background images.

It should be stressed that, usually, the extraction of suitable features (e.g. scale or translation invariant) is considered necessary in image classification, since the pixel values are often very noisy and redundant features that represent images properties in an overly complicated way. However, in Chapter 5 we show that this kind of approach can be successful for particle size classification when the measurement sessions are specifically designed to avoid measurement bias.

In any case, the fact that measurement bias is not detected by background classification does not guarantee that biasing background information is completely removed from the cell holograms. Indeed, the particle presence perturbs the way the background illumination is transmitted to the camera, creating an interference pattern in the acquired image, which is a nonlinear interaction between background and particle information. Therefore, the relation between background hologram h^{bkgr} and cell information C in an acquired cell hologram h^{cell} can be formalized as follows:

$$h^{\text{cell}} = h^{\text{bkgr}} + \mathcal{N}(h^{\text{bkgr}}, C) \tag{2.2}$$

Where $\mathcal{N}$ is an unknown nonlinear function that represents the contribution to the acquired images due to cell presence. Thus we can see that background subtraction can be useful to remove the background information h^{bkgr} that linearly overlaps with the perturbation $\mathcal{N}$ due to the cell presence. However, the potentially biasing background information is also present in the $\mathcal{N}$ contribution.

As a next step, we tried to further improve the classification performance by employing auto-correlation values as features, this time calculated on the background-subtracted 1D holograms. In this case, we considered the number of auto-correlation features (with increasing pixel translation) as a hyperparameter to be optimized through cross validation (see Fig. 2.15).

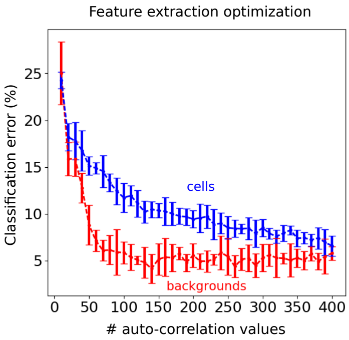


Figure 2.15: Classification error (evaluated through cross validation) as a function of the number of employed auto-correlation features (with increasing pixel translation). The auto-correlation values are calculated from background-subtracted 1D samples. The error bars correspond to twice the standard deviation of the error estimations. The low error of background classification shows that biasing information can be extracted from background samples by the auto-correlation in spite of background subtraction.

We noticed that many more auto-correlation features were required to have a cell classification error below 10%, w.r.t. the previous case without background subtraction. Surprisingly, we also found that the background classification shows quite low error, around 5% (much lower than the random guess error). This means that if the background subtraction removed measurement bias when the classifier was trained with pixel values of background samples, the auto-correlation is nevertheless capable of retrieving the biasing information from these samples.

Therefore, we can conclude that the fast background fluctuations (i.e. faster than the time between measuring the background and the cell) can already provide biasing information. This information can be made available by the use of sensitive enough features, such as auto-correlation values. The main conclusion is that, unfortunately, background subtraction is not even able to remove the biasing linear interaction between background illumination and cell signal. This aspect can be formalized and made clearer by expanding the formalism introduced with Eq. 2.2. In particular, let us consider a cell hologram h_i, where i is the index indicating the chronological order of the holograms acquisition, and a previously acquired background hologram h_{i-1}. We have that the i^{th} background-subtracted hologram $\overline{h}_i$ is given by:

$$\begin{aligned} h_i &= h_i^{\text{bkgr}} + \mathcal{N}(h_i^{\text{bkgr}}, C_i) \\ h_{i-1} &= h_{i-1}^{\text{bkgr}} \\ \overline{h}_i &\equiv h_i - h_{i-1} = \delta h_i^{\text{bkgr}} + \mathcal{N}(h_i^{\text{bkgr}}, C_i) \end{aligned} \tag{2.3}$$

If h_i is a background hologram instead, as it is the case for the samples employed for background classification, the only source of biasing information in background-subtracted samples are given by the fast background fluctuation δh_i^{bkgr}. Auto-correlation was able to retrieve the biasing information from this term.

2.5.2 Feature selection approach

We have seen that the available samples are permeated by biasing information, which does not seem easy to remove. However, we still do not know if this information is well spread over the employed features or if it affects only specific ones. For example, let us consider the frequency components of 1D holograms provided by the absolute value of the FFT. It might be possible that the biasing information, which is the noise component that is correlated with the class labels, is mainly present at certain frequencies. Therefore, if the other non-biasing features were still bringing enough information about the cell-class-dependent characteristics, a bias-free classification could still be possible. In particular, given a certain set of candidate features, we should select the ones that increase the cell classification accuracy without increasing the background classification accuracy as well. In the following approaches we considered an initial set of candidate features composed of auto-correlation and FFT-values calculated on 1D holograms.

Before describing the main machine learning approach we employed, we briefly report other two approaches that we tried without achieving interesting results:

- *Forward feature selection*. Starting from no features, we added one feature at a time to the selected feature set. At each step, we selected the feature to be added that, together with the previously selected ones, provided the best cell classification improvement without improving the background classification as well. In this method, in addition to the logistic regression, a nonlinear classifier (a simple fully-connected neural network with one hidden layer) was tried as well.

- *Modified cost function.* The cost function of the linear classifier (see Chapter 1) was modified so that the learned weights would separate the cell samples and, at the same time, not separate the background samples.

Instead, the main adopted feature selection approach is based on single-feature statistics. Four samples sets were created, each comprising three classes, from which suitable features were selected:

- *Auto-correlation of cell* samples. Classes: monocyte, T cell and granulocyte.
- *Auto-correlation of background* samples. Classes: monocyte background, T cell background and granulocyte background.
- *FFT of cell* samples. Classes: monocyte, t-cel and granulocyte.
- *FFT of background* samples. Classes: monocyte background, T cell background and granulocyte background.

Then we defined a *similarity measure* $J_{A,B}$, which is a measure of how similar the distributions of a given feature are in two sample sets belonging to two classes A and B:

$$J_{A,B} = \sqrt{\frac{1}{2\pi(\sigma_A^2 + \sigma_B^2)}} \exp\left[-\frac{(\mu_A - \mu_B)^2}{2(\sigma_A^2 + \sigma_B^2)}\right] \tag{2.4}$$

Here μ_A and μ_B are the mean values, and σ_A and σ_B the standard deviations, of the given feature over the samples respectively belonging to class A and B. In particular, $J_{A,B}$ is proportional to the probability that the same feature value is sampled from the distributions corresponding to the two classes A and B, assuming that these are normally distributed. More precisely, it corresponds to the probability density of the difference of two normally distributed random variables, imposing that the difference is zero.

Furthermore, we define a similarity measure for three classes A, B and C as:

$$J_{A,B,C} = \sqrt{J_{A,B}^2 + J_{A,C}^2 + J_{B,C}^2} \tag{2.5}$$

We calculated this similarity measure for each feature in the four considered sample sets (see Fig. 2.16).

We can notice that, both for the auto-correlation and the FFT, there are features that separate better the backgrounds and features that separate better the

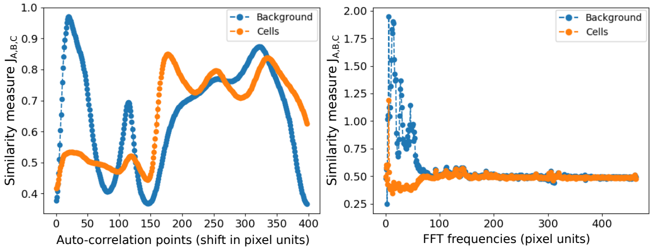


Figure 2.16: Similarity measures for the features in the four considered samples sets: *Auto-correlation of cell* and *Auto-correlation of background* (compared in the first plot), *FFT of cell* and *FFT of background* (compared in the second plot).

cell samples. In particular, the FFT at low frequencies seems to provide interesting features in which backgrounds are very similar but cell samples are not.

As previously mentioned, we aim to select those features that at the same time separate the classes in the cell sample sets well and do not separate the classes in the background sample sets. For a given feature, this requirement can be enforced by considering a suitable measure, which we call *unbiased separation* U, defined as $U = J^{bkgr}/J^{cell}$, where J^{bkgr} and J^{cell} are the similarity measure of a given feature respectively belonging to two corresponding background and cell sample sets. Intuitively, features with high U tend to contain useful (i.e. class-specific) information that is not present in the background images, and that presumably do not originate from the measurement conditions. For a chosen lower threshold value Θ, we selected the features with highest unbiased separation that satisfy the condition $U > \Theta$. In order to find an optimal number of features, we trained and validated the linear classifier for different Θ values (see Fig. 2.17).

Unfortunately, only when we use FFT features and select a small number of them, does the cell classification outperform the background classification. Even using these features, the background classification still shows an error lower than 30%, while around 67% would be expected in case of no bias.

Next, we tried to apply the same machine learning pipeline to background-subtracted 1D holograms (see Fig. 2.18), since we have previously seen that background subtraction can help reducing measurement bias.

Indeed, we can notice that this improved the overall picture. In particular, Only when we use FFT features and select a small number of them, are the classification results close to (but does not reach) the targeted ones, which are good cell classification accuracy (error < 10%) without bias (i.e. background classification error around 67%).

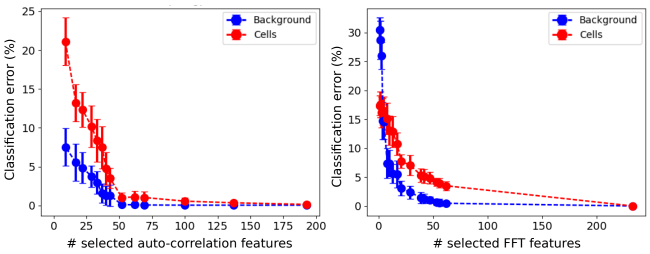


Figure 2.17: Cell and background classification error for different values of the lower threshold Θ applied to the unbiased separation U measure, i.e. for different numbers of selected features (x axis). Auto-correlation (first plot) and FFT (second plot) features are considered. Only when we use FFT features and select a small number of them, is the background classification error higher than the cell classification error, suggesting that those features bring more useful and less biasing information than the others.

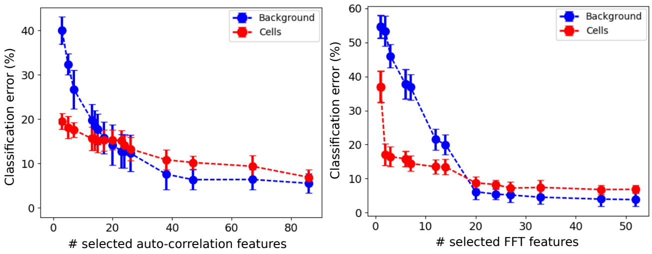


Figure 2.18: Cell and background classification error for different values of the lower threshold Θ applied to the unbiased separation U measure, i.e. for different numbers of selected features (x axis). Auto-correlation (first plot) and FFT (second plot) features calculated on background-subtracted samples are considered. Only when we use FFT features and select a small number of them, are the classification results close to (but do not reach) the targeted ones, which are around 67% background classification error (no bias) and less than 10% cell classification error.

In order to find out if there is a class that is more difficult to distinguish w.r.t. the others, we repeated the same feature selection, classifier training and validation, this time considering three 2-classes problems, each of them throwing away data of one class of the original 3-classes problem. For the sake of brevity, we do not present the outcome in detail. From the obtained results using background-subtracted samples it became clear that the *monocytes* v.s. *granulocytes* classification is by far the most challenging, but also the least biased, showing higher error for both background and cell samples. It should be noticed that we did not encounter this strong class unbalance before, e.g. in the previously presented confusion matrices.

Looking back at the average background illumination profiles of the different measurement sessions (see Fig. 2.9), we might suspect that these class-unbalanced performances could be caused by the significant deformation of the background illumination in the T cell measurement session. In order to remove this possibility and momentarily simplify our attempt to demonstrate an unbiased cell classification, in the next subsection we will only focus on the *monocytes* v.s. *granulocytes* classification.

2.5.3 *Beneficial confusion* approach

In light of the results so far, we know that the fast background fluctuations δh^{bkgr} (see Eq. 2.3) can deliver biasing information to the classifier, and that background subtraction and feature selection (selecting FFT and auto-correlation values) could not remove such a disruptive influence. However, when cell classification is concerned, we still do not know how much the nonlinear term $\mathcal{N}(h^{\text{bkgr}}, C)$ contributes to the observed measurement bias. In this subsection we will remove the bias caused by the linear term h^{bkgr} and shed light on this aspect, by seeing how much nonlinear bias still remains.

We employ a method, which we call *beneficial confusion* (BC), that allows to prevent the measurement bias due to the linear interaction between background and cell signal. This consists in adding to each hologram a randomly selected background hologram from another measurement session. Therefore, considering two measurement sessions A and B (corresponding to the monocyte and the granulocyte sessions in our case) we compose new sample sets of *BC holograms* $\widetilde{h}$ by combining the original samples in the following way (we employ the formalism of Eq. 2.2):

$$\begin{aligned}
\widetilde{h}^{\text{cell}}_{i,A} &\equiv h^{\text{cell}}_{i,A} + h^{\text{bkgr}}_{j_i,B} = h^{\text{bkgr}}_{i,A} + \mathcal{N}(h^{\text{bkgr}}_{i,A}, C_{i,A}) + h^{\text{bkgr}}_{j_i,B} \\
\widetilde{h}^{\text{bkgr}}_{i,A} &\equiv h^{\text{bkgr}}_{i,A} + h^{\text{bkgr}}_{j_i,B} \\
\widetilde{h}^{\text{cell}}_{i,B} &\equiv h^{\text{cell}}_{i,B} + h^{\text{bkgr}}_{j_i,A} = h^{\text{bkgr}}_{i,B} + \mathcal{N}(h^{\text{bkgr}}_{i,B}, C_{i,B}) + h^{\text{bkgr}}_{j_i,A} \\
\widetilde{h}^{\text{bkgr}}_{i,B} &\equiv h^{\text{bkgr}}_{i,B} + h^{\text{bkgr}}_{j_i,A}
\end{aligned} \tag{2.6}$$

where the sample index i follows the chronological order of the image acquisition, while the sample index j_i does not: it is a random permutation (shuffling) of the chronological index i. It can be noticed that these artificial samples always present a linear overlap of both the background information from the two measurement sessions A and B. Thus, the corresponding biasing information is counterbalanced, and the classifier is in a way “confused”, so that the linear contribution of h^{bkgr} cannot be exploited for the classification training.

In order to provide an intuitive understanding of how and why the BC method would reduce measurement bias, we propose the following analogy. Let us imagine that we want to train a classifier to distinguish whether in a picture there is a crow or a seabird. To do so, we build our training sample set by automatically retrieving a large number of related pictures from the web. It is obvious to us as humans that the proper way to perform the classification is to detect the position of the bird in the picture and look at its properties, such as colour and shape. Thus we might expect that, once the classifier shows a good capacity of classifying the images, it learnt to correctly classify the birds by performing similar operations. But we might be wrong.

Indeed, if there is any other available way to predict the correct labels of the images, which from the classifier's perspective is easier to learn, the training algorithm will most likely learn that. In our case, let us suppose (quite realistically) that most of the crow images presents some part of a tree or grass in the background, while most of the seabird pictures show some view of a sea or beach. It is then plausible that the classifier has learned the easier task of distinguishing woods or meadows from seaside in the pictures. This would be an example of data set bias or shortcut learning [6] and a case similar to the measurement bias we have investigated. To check if this specific shortcut learning has happened, we can, just as we did before when trying to classify backgrounds without cells, try to submit wood, meadow and seaside pictures without birds to the trained classifier. If the training was correctly focused on the bird, the classifier should not be able to distinguish these background images (i.e. should show around 50% accuracy). Otherwise, we have proof that shortcut learning has happened.

Now, let us see how we can use the BC method to try to solve this problem without looking for more suitable pictures for training. We can, for example, program an algorithm that copies part of the background from one class of pictures and paste it to part of the background of the other class pictures, and vice versa, without covering the bird shape. The result would be that, approximately, all the pictures have now both tree parts (or meadow) and sea (or beach) in the background. Therefore, by considering these BC samples, the training algorithm would be "beneficially confused" and unable to use background information to classify the pictures. It would instead be obliged to employ the bird information, since there is no "easier" way left. Hopefully, the learned classification can be successfully applied to the original pictures as well, since the background should not be very relevant anymore.

Going back to our cell classification problem and to the BC samples composed as shown in Eqs. 2.6, we now expect that background classification cannot be learned anymore and that only the term $\mathcal{N}$ can be exploited for learning cell classification. Indeed, employing the feature selection approach described in the previous subsection and considering FFT values of BC 1D holograms, we obtain around 50% error in background classification and less than 10% error in cell classification (see Fig. 2.19). This result implies that the information present

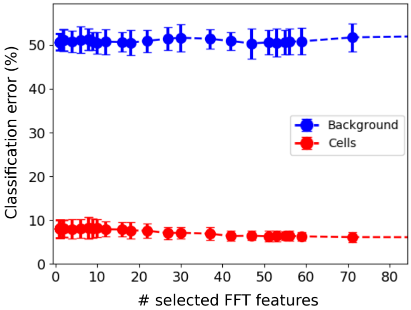


Figure 2.19: Cell and background classification error (only 2 classes considered: monocytes vs. granulocytes) for different values of the lower threshold Θ applied to the unbiased separation U measure, i.e. for different numbers of selected features (x axis). FFT values of beneficial confusion (BC, see eq. 2.6) 1D holograms are considered as features. As expected, the BC method makes background classification impossible (around 50% error), while cell classification shows good accuracy (< 10%).

in the hologram perturbation $\mathcal{N}(h^{\text{bkgr}}, C)$ due to the cell presence is sufficient to classify the samples with good accuracy. It should be noticed that good accuracy is achieved by selecting only a single FFT feature. This means that there is a single frequency component that alone can separate most of the samples according to the two classes. We will look into this detail further on.

For now we are still left with two important questions:

1. We do not know how much of the obtained accuracy is due to the contribution of the background biasing information h^{bkgr} in the nonlinear interaction $\mathcal{N}$ ad how much is originated by the actual cell type characteristics C.

2. We do not know if the classifier trained using BC samples can be directly applied to successfully classify the original samples as well.

For now, let us try to find an answer to the (easier) second question. As we will see, this will lead to an answer to the first question as well. Of course, the answer to the second question depends on the type of extracted features. In particular, if the features are extracted through a linear function of the pixel values of the hologram, the weights of the linear classifier learned using BC holograms might provide similar classification accuracy when applied to the original holograms, provided that the intercept (i.e. a constant that is added to the weighted sum of the logistic regression, see Chapter 1) is properly adapted.

An intuitive motivation for this statement can be given by considering cell BC samples as described in Eqs. 2.6. Once the weights w_k (where k is the index for the employed features) and the intercept w_0 of the linear classifier are trained using BC samples, the condition for BC samples $\widetilde{h}_{i,A}^{\text{cell}}$ and $\widetilde{h}_{i,B}^{\text{cell}}$ to be correctly classified is:

$$\sum_k w_k f(\widetilde{h}_{i,A}^{\text{cell}})_k \geq -w_0$$
$$\sum_k w_k f(\widetilde{h}_{i,B}^{\text{cell}})_k < -w_0$$

I.e. the weighted sum of the extracted features $f(\widetilde{h}_{i,A}^{\text{cell}})_k$ and $f(\widetilde{h}_{i,B}^{\text{cell}})_k$ are separated by the threshold value $-w_0$. Let us assume that the learned BC sample classification performs well, as it does in our case (see Fig. 2.19), and that therefore these inequalities are true for most samples.

If the function f that is used to extract the features is a linear function, considering the relations in Eqs. 2.6, the inequalities can be written as:

$$\sum_k w_k f(h_{i,A}^{\text{cell}})_k \geq -w_0 - \sum_k w_k f(h_{j_i,B}^{\text{bkgr}})_k$$
$$\sum_k w_k f(h_{i,B}^{\text{cell}})_k < -w_0 - \sum_k w_k f(h_{j_i,A}^{\text{bkgr}})_k$$

We can notice that these are in turn the conditions for the original cell samples $h_{i,A}^{\text{cell}}$ and $h_{i,B}^{\text{cell}}$ to be correctly classified by the linear classifier trained with BC samples but using a modified intercept, provided that the right hand sides of the two inequalities are sufficiently similar in average value.

Instead of proving this last condition, we directly tried the original sample classification using many different intercept values. In particular, we selected the FFT feature set (55 values) that scored best in the classification trained and validated on BC cell holograms (Fig. 2.19, red points). Then we applied the trained classifier to the original 1D holograms test sets (employing the selected features) to evaluate the classification performance, for different values of the intercept. We obtained a cell classification error below 10% (see Fig. 2.20 a), proving that the classifier trained using BC samples could be successfully applied to the original samples.

However, it should be noticed that the employed FFT features are obtained by calculating the absolute value of the FFT of 1D holograms, which is not a linear function due to the absolute value operation. Nevertheless, the approach still works. Indeed, it is likely that the specific combination of frequency components useful to classify the BC hologram, is also useful (at least partially) to classify the original holograms. We therefore have answered the second question.

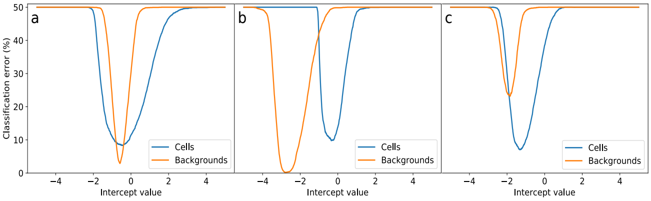


Figure 2.20: Cell and background classification error, obtained by applying the linear classifier trained using BC cell samples, as a function of the intercept value of the linear classifier. *a*: The employed FFT feature set (55 values) is the one that scored best in the classification trained and validated on BC cell holograms (Fig. 2.19, red points). *b*: The single FFT feature that provided the highest BC cell classification accuracy was considered, that is the 5^{th} frequency component value. *c*: The employed FFT feature set (66 values) is the one that scored best in the classification trained and validated on BC cell holograms, when the best FFT feature (i.e. the 5^{th} frequency component) was excluded from the selection. These three plots show that the classifier trained using BC cell samples could be successfully applied both on the original cell and background samples. Therefore, the training was affected by measurement bias.

However, we notice that also the background images could be (even more) successfully classified employing the classifier trained with the BC cell samples. This suggests that FFT can retrieve biasing background information from the nonlinear background-cell interaction $\mathcal{N}(h^{\text{bkgr}}, C)$, and that this information can be used to classify background images as well. Therefore, this seems to answer our first question.

In order to make sure that the high accuracy shown by cell classification and background classification is actually due to the same information, we repeated the test classification for different intercept values, this time selecting only the single best FFT features. Indeed, we previously noticed that one frequency component (which is the 5^{th} FFT value, that is rather high frequency) is responsible for most of the accuracy in the BC sample classification (Fig. 2.19). As expected, employing the same selected frequency component, both cell and background classification still show low error (Fig. 2.20 b). This confirms that the classifier has learnt to use a frequency component which is useful to separate backgrounds instead of cell classes. Moreover, if we remove the 5^{th} FFT value from the features available for selection and we employ the best feature set found (66 values), we still obtain high test accuracy in cell classification, but significantly lower test accuracy in background classification (see Fig. 2.20 c). However, the background test accuracy is still significantly higher than 50%, implying that biasing information could be extracted from $\mathcal{N}(h^{\text{bkgr}}, C)$ also through other frequency components.

We have therefore seen that cell perturbations $\mathcal{N}$ in the acquired holograms can easily convey biasing information to the training algorithm, preventing us from developing and evaluating a trusted machine learning approach. Unfortunately, the nonlinear interaction between background and cell information prevents us from removing measurement bias using background subtraction or the proposed BC method. Thus, we conclude that, in the case of cell hologram classification, it is very difficult to prevent measurement bias through image post-processing or validation methods. Instead, the bias problem should be dealt with by adopting a suitable configuration of measurement sessions to generate samples that are general enough, in order to ensure that generalization over different measurement conditions can be achieved and demonstrated (this is done in Chapter 5).

Another conclusion is that the proposed BC method is more effective in treating measurement bias w.r.t. background subtraction. For example, such a method can be useful when, for each class, at least the test sample set is measured under conditions uncorrelated to the ones of the measurement of training samples, enabling a proper validation of the learned classification.

2.6 Summary and conclusion

In this chapter we presented an investigation based on machine learning experiments aimed to develop an effective and computationally efficient algorithm to classify three types of white blood cells, employing the holograms acquired in a lens-free and label-free imaging microflow cytometer.

At first we have studied the main properties of the over 50,000 hologram images provided by our project partners from IMEC. In particular, we have discussed and quantified the main sources of sample variability due to changes of background illumination during the measurements. We have found that these fluctuations are significant and may affect different properties of the acquired background holograms, such as shape, intensity and position.

In this work we chose to greatly diminish sample dimensionality by summing the pixel values of the images along the channel direction, to obtain 1D holograms. This operation reduces the noise due to cell hologram displacement along the channel direction in the images. Moreover, it approximates the sample acquisition of a line-scan image sensor, which has the advantage of a much higher frame rate w.r.t. cameras. Mainly, the auto-correlation and fast Fourier transform (FFT) values calculated on the 1D holograms were considered as features to train a linear classifier, because of their sensitivity and robustness against sources of noise in the samples, such as cell hologram displacement and fluctuations in background intensity and position. Furthermore, we developed an algorithm that could automatically and accurately distinguish background samples from cell samples, based on a moving threshold applied to the sum of suitable auto-correlation values.

Although we could easily obtain high accuracy in the classification of cell samples, we discovered that this was true for the classification of background samples as well. This was possible because the samples were acquired in a single measurement session for each cell type, and therefore the samples belonging to different classes were distinguishable by specific background illumination properties due to drifts in measurement conditions. These circumstances could cause the training algorithm to focus on biasing background information when trying to solve the classification problem, instead of employing the correct class-specific information provided by the cell properties. Such a bias of the learning process, which we called *measurement bias*, is an example of the well-known machine learning issue generally referred to as *data set bias* or *shortcut learning*, which is often underestimated or ignored in works proposing new machine learning applications.

Therefore, our aim shifted to a better understanding of the measurement bias affecting our classifier, in order to achieve bias-free classification. In the case of cell holograms, biasing background information can be accessed by our classifier from two distinguishable sources in the samples: the linear and the nonlinear components of background-cell interaction. The linear component can be partially removed by background subtraction, but we proved that this is insufficient: the auto-correlation function can still retrieve the biasing information from the linear background component in the samples.

Afterwards, we implemented and tested a feature-selection method that aims to select useful but non-biasing features (such as specific FFT frequency components), i.e. which well separate the cell holograms according to the classes, but not the background holograms according to the measurement session. Even though this approach provided interesting insight into the problem, we could not completely separate biasing from useful information and find a definitive solution.

We then developed a method, called *beneficial confusion* (BC), which allows to completely remove the measurement bias due to the linear background-cell interaction. By means of this we could finally demonstrate that significant biasing information is still conveyed by the nonlinear component of the cell-background interaction. Because of nonlinearity, such a source of measurement bias is difficult to remove, or even to assess, through traditional techniques such as background subtraction, cross validation and feature selection, as we contributed to demonstrate. Instead, the key approach to address this issue lies in performing suitable measurement sessions, which should be designed to avoid correlation between class labels and measurement conditions. These conclusions generally apply to imaging microflow cytometry applications where accurate ground truth (i.e. class labels in classification tasks) cannot be determined within the same citometry measurement which generates the samples.

In conclusion, we should stress that at least two different measurement strategies can be employed to treat measurement bias. The one that requires least measurement-related effort consists in acquiring training samples and test samples in separated measurement sessions for each class. In this case, the aim is to obtain test samples that, with reference to the training samples, sufficiently represent the general changes in measurement conditions that would occur in the real-life usage of the cytometer. I.e., the main requirement is that the correlation between class labels and measurement conditions that characterize the training samples is completely broken in the test sample measurements. This at least provides the possibility to check whether and to which degree the employed classifier was affected by measurement bias. When this approach is considered, the investigation and the methods that we presented in this chapter can be helpful in reducing measurement bias through post-processing (software based) solutions.

A more complete and effective approach is to make sure that also the training samples are acquired through measurement sessions that break the correlation between class labels and the effects of measurement conditions. This allows to prevent measurement bias at the root, since it directly removes the possibility that the classifier could learn the classification task through measurement conditions. In this case, if a proper validation technique is employed, there is no need to apply dedicated software-based techniques against measurement bias. In Chapter 5, we indeed explore this strategy in an ad-hoc experimental demonstration.

References

[1] Liesbet Lagae, Dries Vercruysse, Alexandra Dusa, Chengxun Liu, Koen de Wijs, Richard Stahl, Geert Vanmeerbeeck, Bivragh Majeed, Yi Li, and Peter Peumans. *High throughput cell sorter based on lensfree imaging of cells.* In 2015 IEEE International Electron Devices Meeting (IEDM), pages 13–3. IEEE, 2015.

[2] Yuqian Li, Bruno Cornelis, Alexandra Dusa, Geert Vanmeerbeeck, Dries Vercruysse, Erik Sohn, Kamil Blaszkiewicz, Dimiter Prodanov, Peter Schelkens, and Liesbet Lagae. *Accurate label-free 3-part leukocyte recognition with single cell lens-free imaging flow cytometry.* Computers in biology and medicine, 96:147–156, 2018.

[3] Bruno Cornelis, David Blinder, Bart Jansen, Liesbet Lagae, and Peter Schelkens. *Fast and robust Fourier domain-based classification for on-chip lens-free flow cytometry.* Optics Express, 26(11):14329–14339, 2018.

[4] Bendix Schneider, Joni Dambre, and Peter Bienstman. *Fast particle characterization using digital holography and neural networks.* Applied optics, 55(1):133–139, 2016.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[6] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. *Shortcut learning in deep neural networks.* Nature Machine Intelligence, 2(11):665–673, 2020.

3

Cell classification improved by on-chip dielectric scatterers: a numerical investigation

In Chapter 2 we presented a series of machine learning experiments that aimed to efficiently classify three different white blood cell (WBC) types employing an imaging label-free microflow cytometer. In that case, the employed samples were the interference patterns (or holograms) projected on the image sensor by a laser beam which passes through a microfluidic channel and a flow cell. Originally, this system was meant to perform inline digital holographic microscopy, and therefore to enable software-based reconstruction of cell images from the acquired holograms [1]. However, when fast classification is targeted, e.g. for online operations in high-throughput cytometry such as cell sorting, image reconstruction algorithms are usually too computationally expensive and it is therefore advantageous to bypass them [2]. Nevertheless, also powerful machine learning algorithms for image classification, e.g. based on feature extraction or on convolutional neural networks, might take too long to run when throughputs higher than around 1000 cells/s are targeted [3–6]. Therefore, in this case, it is greatly desirable to perform part of the required classification operations directly in the optical domain, before the image is acquired by the camera. In this chapter we will explore such an approach by means of finite-difference time-domain (FDTD) optical simulations [7]. In particular, for cell classification, we employ the extreme learning machine (ELM) approach (see Chapter 1), which is a powerful machine-learning framework that is particularly suitable for simple and computationally cheap hardware implementations.

The main idea consists in interposing a suitable diffractive layer between the microfluidic channel of the cytometer and the camera that acquires the interference patterns. The function of the diffractive layer is, intuitively, to better exploit the interference effects enabled by the use of coherent light in order to enrich the information content of the measured patterns. In more specific terms, we aim to increase the dimensionality of the cell information represented in the acquired interference pattern. Indeed, the main concept behind the ELM approach is that the performance of a simple linear classifier (Ref to ML intro) can be enhanced if applied to a higher-dimensional, random and nonlinear representation of the input data. Our implementation is enabled by the fact that the image sensor performs a nonlinear operation on the optical phase conveying the cell information, by transforming it into a light intensity pattern. The aim of our FDTD simulations is to provide a proof-of-principle demonstration that a diffractive layer can be used to improve cell classification through the technique we propose.

This chapter is an extension of the related work published in [6, 8, 9].

3.1 The main concept: dielectric scatterers to improve fast cell classification

In this work we consider a passive, linear, integrated photonic stage as an interface between the cell illumination stage and the image sensor. The goal is to simplify and speed up the classification process in the slow software domain. In particular, the laser beam that passes through the microfluidic channel is made to propagate across a collection of silica pillar scatterers of elliptic cross section embedded in silicon nitride (Fig. 3.1). For computational efficiency reasons, this process has been investigated via 2D FDTD simulations as a proof-of-concept, approximating the 3D case of a cell flowing in a microfluidic channel interfaced with a photonic chip. The far-field intensity of the light exiting the scatterers cluster is collected by a 1D array of virtual pixels that approximately simulate a line scan image sensor. Indeed, in the previous chapter we have seen that working with 1D images can be beneficial for high speed machine learning classification of cells, for several reasons. Finally, the pixel outputs are fed into a linear classifier that can be, for example, implemented in software.

The photonic stage containing the scatterers is intended to exploit the nonlinearity of the transfer function that relates the phase shift accumulated by the light through the cell to the corresponding interference pattern measured by an image sensor. As previously mentioned, the system can be seen as an hardware implementation of an ELM (see Chapter 1 and Subsection 3.4.2 for a further argumentation). Generally, the main advantages of ELM with respect to other machine learning techniques are that only a linear readout (in this case a linear classifier) needs to be trained and that it is easily implemented in hardware. In this case, the pillar scatterer stage determines the ELM hidden node structure by projecting onto the far-field intensity a very intricate nonlinear mapping, based on sinusoidal functions of the phase information. This parallel processing is carried out nearly instantaneously with respect to both the cell movement and the operating speed of an electronic computer, providing an important advantage over other software-based machine learning solutions.

It should be stressed that the phase-to-intensity nonlinearity is already expressed by the interference pattern projected by the cell alone, without scatterers. However, we aim to demonstrate that the dimensionality of such a nonlinear mapping can be enhanced by the use of scatterers in order to increase the performance of a subsequent linear classification. The exploitation of light interference in order to enable a passive integrated reservoir computing implementation (which is based on similar principles of ELMs but is applied to time-dependent signals) using linear optical media was demonstrated in [10]. However, the time-dependent input information was encoded in the intensity of a laser signal and therefore the transfer function to the output detector was quadratic (amplitude to intensity). Here, the input information is encoded in the phase of a laser signal, such that the readout transfer function can be sinusoidal with respect to that input (see Subsection 3.4.2, for a further argumentation). The sine, for example, can be profitably employed as activation function in feedforward neural networks under suitable conditions [11].

In summary, we aim to employ FDTD simulations to investigate how the classification of cells performs when dielectric scatterers in different configurations and conditions are used, considering as a baseline the case without any scatterer.

3.2 Simulation details

Before going into detail, it should be stressed that each 2D FDTD simulation is meant to generate a sample, i.e. the 1D interference pattern projected by the simulated cell and by the dielectric scatterers. For each cell class we need to run several (thousands) simulations with the same scatterer configuration but different cell conditions (e.g. given by the cell position, rotation, etc...), which provides the main source of variability in the training of the classifier. Additionally, white noise is added to each acquired cell hologram, accounting for the non-ideality of image acquisition.

The FDTD method was chosen because of its reliability in dealing with complex dielectric structures (provided that a fine enough mesh size is chosen) and because it allows for an intuitive understanding of the computed physical pro-

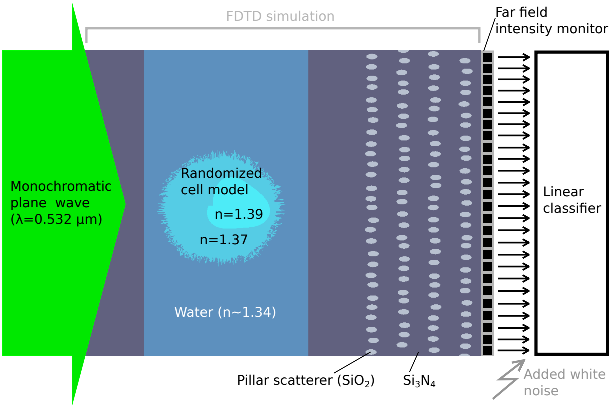


Figure 3.1: Schematic of the classification pipeline, including an example of 2D FDTD simulation. A monochromatic plane wave impinges on a microfluidic channel containing a randomized cell model in water ($n_{H_2O} \approx 1.34$, $n_{\text{cytoplasm}} = 1.37$, $n_{\text{nucleus}} = 1.39$). The forward scattered light passes through a collection of silica scatterers ($n_{SiO_2} \approx 1.461$) embedded in silicon nitride ($n_{Si_3N_4} \approx 2.027$) and organized in layers. The radiation intensity is then collected by a 1D far-field monitor, which is divided into bins (pixels). Each pixel value is fed into a trained linear classifier (logistic regression) that consists of a weighted sum of the pixel values.

cess. In particular, visualizing the light field as a function of time provided a qualitative idea of how and how much the light signal coming from the cell was mixed by the dielectric scatterers.

Both a cell in a microfluidic channel and the proposed silica scatterers were modeled in the same 2D FDTD simulation (Fig. 3.1) employing Lumerical's FDTD Solutions software. A monochromatic plane wave (vacuum wavelength $\lambda = 532$ nm, chosen as in [12]) of constant intensity impinges transversely on a microfluidic channel (15 μm wide) filled with water. The channel interfaces with a region containing layers of elliptic scatterers (0.5 μm wide and 1 μm long) made of SiO_2 embedded in Si_3N_4. At the end of the scatterer region, a vertical far-field monitor (with an angular resolution of 55.6 points/deg) covers the total length of the simulated space. The simulation region is 28 μm long along the vertical direction and from 20 μm to 30 μm along the horizontal direction, depending on the number of scatterer layers. The FDTD mesh size is $\approx \lambda/29$.

It should be stressed that the scope of this work is to provide a proof-of-concept of a new approach to photonics machine learning, that can be generalized to many other implementations besides the discussed examples. Therefore, the dimensionality of the simulation, the size of the structures and the cell model are a consequence of a trade-off between closeness to reality, saving of computational time and the search of a sufficiently complex (but not overly so) task. Indeed, since we need to run thousands of FDTD simulations to generate enough samples for a single classification task, keeping the computational cost low is a key requirement for the feasibility of this study. In any case, all the simulated objects, aside from the scatterers, were designed beforehand and independently of the classification results.

3.2.1 Randomized cell model

The cell model is composed of a cytoplasm region ($n_{\text{cytoplasm}} = 1.37$) surrounding a nucleus region ($n_{\text{nucleus}} = 1.39$). An example is shown in Fig. 3.1. In order to generate a different cell shape for each simulation, a randomized 2D cell model was employed, based on distorted ellipses. Considering the ellipse equation in polar coordinates (ρ is the distance from the origin and θ the angle with respect to the horizontal axis)

$$\frac{\rho^2 \cos^2 \theta}{a^2} + \frac{\rho^2 \sin^2 \theta}{b^2} = 1 \tag{3.1}$$

a surface modulation is introduced through the following substitution:

$$\rho \rightarrow \rho \left(1 + A\cos(\omega\theta)\right) \tag{3.2}$$

In addition, irregularities of the cell external surface are simulated by adding a noisy high-frequency modulation through:

$$\rho \rightarrow \rho + B\varepsilon_s \tag{3.3}$$

where ε_s is a random number sampled from an uniform distribution from -1 to +1 for each point on the surface. The cytoplasm and the nucleus were modelled using a polygon of 1000 vertices, thus the last substitution introduces 1000 additional random variables in the cell model.

In this chapter, two different classification tasks are considered. The first is based on average nucleus size and aims to distinguish between "normal" cells (small nucleus) and "cancer" cells (bigger nucleus). The names in quotation marks were chosen because of the common tendency of cancer cells to show evident irregularities in nucleus size [13]. The second task is based on nucleus shape and aims to distinguish between "lymphocytes" (big quasi-spherical nucleus) and "neutrophils" (nucleus divided in 3 lobes). The names in quotation marks refer to two among the most common white blood cells that are present in human blood. These two cell models are, physically and biologically speaking, only rough representations of their real counterparts when flowing in a liquid medium [14]. The scope of this work, though, is to demonstrate the differences between two classification approaches applied on models that are sufficiently representative of real cell cases from a machine-learning perspective.

"Normal" and "cancer" cells

The parameters for "normal" and "cancer" cell models are chosen as follows (subscript c stays for "cytoplasm" and n for "nucleus"): $a_c = b_c = 5 \times (1 + 0.1\varepsilon)$µm, $A_c = 0.1 \times (1 + 0.9\varepsilon)$, $\omega_c = 3 \times (1 + 0.9\varepsilon)\text{rad}^{-1}$, $B_c = 0.2$ for cytoplasm and

$$a_n = b_n = \begin{cases} 1.2 \times (1 + 0.1\varepsilon)\text{µm} & \text{“normal” cells} \\ 2.5 \times (1 + 0.1\varepsilon)\text{µm} & \text{“cancer” cells} \end{cases} \tag{3.4}$$

$A_n = 0.1 \times (1 + 0.9\varepsilon)$, $\omega_n = 3 \times (1 + 0.9\varepsilon)\text{rad}^{-1}$, $B_n = 0$ for nucleus. Here ε is a random variable with uniform distribution from -1 to +1. In addition, the cytoplasm and the nucleus center displacements are given respectively by $x_c = y_c = \varepsilon$ µm, $x_n = x_c + a_c \times 0.3\varepsilon$ and $y_n = y_c + b_c \times 0.3\varepsilon$. Note that even if the expressions for the couples (a_c, b_c) , (a_n, b_n) and (x_c, y_c) are equal, they can differ in their values being ε a random variable.

In order to provide the reader with an intuitive idea of the shapes and the randomness of the employed cell models, some examples are represented in Fig. 3.2 a.

"Lymphocytes" and "neutrophils"

The main difference between the "lymphocyte" and the "neutrophil" models is that the first has a big quasi-spherical nucleus while the latter has three quasi-spherical nuclei whose average total area is equal to the average area of the "lymphocyte" nucleus. The corresponding parameters are chosen as follows (subscript c stays for "cytoplasm" and n for "nucleus"): $a_c = b_c = 5.5 \times (1 + 0.1\varepsilon)$µm, $A_c = 0.03 \times (1 + 0.67\varepsilon)$, $\omega_c = 3 \times (1 + 0.9\varepsilon)\text{rad}^{-1}$, $B_c = 0.2$ for cytoplasm and

$$a_n = b_n = \begin{cases} 3.5 \times (1 + 0.2\varepsilon)\text{µm} & \text{“lymphocytes”} \\ 3.5/\sqrt{3} \times (1 + 0.2\varepsilon)\text{µm} & \text{“neutrophils”} \end{cases}$$

$A_n = 0.025 \times (1 + \varepsilon)$, $\omega_n = 4.5 \times (1 + 0.33\varepsilon)\text{rad}^{-1}$, $B_n = 0$ for nucleus. Also here, ε is a random variable with uniform distribution from -1 to +1. The cytoplasm center displacements is $x_c = y_c = \varepsilon$. The nuclei center displacements are

$$\begin{array}{ll} x_n = x_c + a_c \times 0.2\varepsilon \text{ and } y_n = y_c + b_c \times 0.2\varepsilon & \text{“lymphocytes”} \\ x_n = x_c + r\cos(\alpha_k + \beta) \text{ and } y_n = y_c + r\sin(\alpha_k + \beta) & \text{“neutrophils”} \end{array}$$

Where $r = 2.75 \times (1 + 0.2\varepsilon)$µm, $\beta = (1 + \varepsilon)\pi$ and $\alpha_k = \frac{2}{3}\pi k + \frac{\pi}{12}\varepsilon$, being $k = 0, 1, 2$ a different integer for the 3 nuclei in a "neutrophil". Some examples are represented in Fig. 3.2 b.

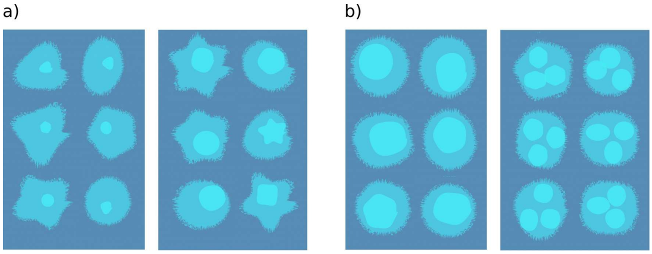


Figure 3.2: Examples of cells automatically generated by the employed randomized models. **a)** Comparison between generated examples of "normal" cell and "cancer" cell. **b)** Comparison between generated examples of "lymphocyte" and "neutrophil".

3.2.2 Scatterer configuration

The scatterer configuration has a large number of degrees of freedom and its complete exploration would be computationally quite expensive. In fact, for each tested configuration, hundreds or thousands of simulations would have to be performed in order to provide the classifier with a sufficient number of training and test samples. Therefore, only a few general parameters that control the complexity of the collected interference pattern were explored, looking for a maximum in the classifier accuracy.

The scatterers are placed in vertical layers (e.g. see Fig. 3.1) with an average vertical distance of $1\mu\text{m}$ between their centers. The center of each scatterer is randomly displaced with respect to their unperturbed center in the layer, both along the vertical and the horizontal directions. All the random displacements belonging to the same architecture are sampled from the same uniform probability distribution. The considered parameters for the structure optimization are:

- The random displacement amplitude A_r of the scatterers, chosen among $A_r = 50\text{nm}, 100\text{nm}, 150\text{nm}, 200\text{nm}, 250\text{nm}$.
- The horizontal distance D between the layers, chosen among $D = 1.846\mu\text{m}, 2.85\mu\text{m}, 3.40\mu\text{m}$. The two last values respectively give a maximum and a minimum in the far-field transmission, considering four layers of scatterers without any random displacement.
- The number N_{layers} of layers, chosen among $N_{\text{layers}} = 1, 2, 3, 4, 5$.

The parameter combination that provided the best accuracy in the classification based on the nucleus size was chosen. An example of how these parameters can modify the interference pattern is given in Fig. 3.3. For the chosen configuration, additional samples were generated and the results provided in the next sections were obtained.

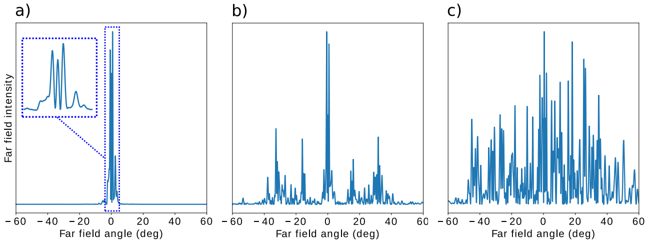


Figure 3.3: Far-field intensity profiles of the light scattered by a cell: **a)** without the presence of scatterers, the interference pattern is relatively simple and smooth, most of the intensity is confined between $-6°$ and $6°$; **b)** with 1 layer of scatterers, the far-field intensity is distributed around periodically placed peaks, most of the field stays between $-40°$ and $40°$; **c)** with 4 layers of scatterers, the far-field intensity is distributed in a complex pattern mostly between $-60°$ and $60°$.

In the first part of the work presented in this chapter, the evaluation of the classification accuracy for different combinations of the scatterers parameters was performed using a relatively low number of samples, i.e. 800 (400 per class). This choice was made because the learning curve (see Chapter 1), calculated using an initial guess for the scatterer parameters (that is $N_{\text{layers}} = 4$, $A_r = 100\text{nm}$ and $D = 1.846\mu\text{m}$), converged for 800 samples or more. However, this way of choosing the number of samples employed for the scatterers optimization is in principle not completely correct, and can only provide a very rough estimation of the classification performance due to different scatterer configurations. Indeed, each scatterer configuration might have different minimum number of samples required to avoid most of the overfitting. This information should be obtained by considering a learning curve for each scatterer configuration. In Subsection 3.5.3 in this chapter we present a more rigorous and robust exploration of the classification performances for different scatterer configurations. In any case, in agreement with those results, most of the investigated configurations provided a similar accuracy.

3.3 Machine learning aspects

In this section we describe the machine learning details and pipeline employed in the next sections to obtain the presented classification results (when not specified otherwise).

Approximating the operation of line scan sensor, the far-field intensity profile was divided into N_{pix} bins (or pixels) and the integration over each bin was fed into a linear classifier based on the logistic regression (see Chapter 1). The Scikit-learn Python library [15] was employed, using the "liblinear" solver. For each tested scatterer configuration a number N_{samp} of simulations was performed randomly varying the cell shape and position, as described in the previous section. In particular, the classification results reported in the next sections are obtained from sample sets of $N_{\text{samp}} = 3200$ samples each (if not specified otherwise). In half of the N_{samp} simulations a "normal" cell was considered, while in the remaining half a "cancer" cell (with bigger nucleus) was used. 75% of these two sets was employed in the training of the logistic regression, while the rest was used as a test set.

Gaussian white noise was added a posteriori to the interference patterns before they were divided into bins. Different values for the noise standard deviation were used: 1%, 5%, 10%, 20%, 30%, 50% and 100% of the average intensity over the sample set.

A study on the dependence of the classification test error on the regularization strength (L1 and L2) and on N_{samp} was carried out in the two cases with and without the use of scatterers, on a set of 2000 samples each. For this investigation, a 4-layer scatterer configuration was considered, with $A_r = 150$nm and $D = 1.846$µm (as previously mentioned, this particular configuration was one of the best performing in a preliminary exploration based on 800 samples). The study pointed out that regularization had no significant positive effect on the performance of the two classification tasks. Furthermore, it showed that the learning curve (test error vs. number of samples) converged around $N_{\text{samp}} = 800$ when the scatterers were used. Thus, for the classifications presented in the next section no regularization was considered. However, we monitored the classification accuracy for different levels of white noise added to the acquired cell holograms, which has an effect similar to employing regularization with different strengths (see Chapter 1 and [16]). The dependence of the classification performance on the number of pixels was kept under direct control by performing sweeps for each tested configuration.

If not specified otherwise, the presented results were obtained through a validation process in which the simulated data samples were randomly shuffled before they are split into training and test sets. From the results generated by repeating this procedure 20 times a mean value and a confidence interval (chosen as $\pm$ twice the standard deviation) were calculated and plotted. Note that a different noise vector is added to the intensity profiles after each shuffling.

3.4 Cell classification improvement due to the dielectric scatterers

In this section we investigate how and in which cases the performance of the simulated cell classification (based on the nucleus size, see Subsection 3.2.1 and Fig. 3.2 a) is improved by the use of the dielectric scatterers, considering as a baseline the case without scatterers. Moreover, the link between the considered physical system and an ELM network is discussed, providing insight on how the computational power of the employed classification system can be improved. In particular, two ways of enhancing the classification performance are numerically demonstrated, which respectively consist in employing a UV laser and in enclosing the channel and cell in an optical cavity.

3.4.1 Green laser light

Let us consider a green laser source ($\lambda = 532$nm) and let us compare the classification error on the test samples when no scatterers are present and when, instead, 4 scatterer layers are employed (considering the random displacement amplitude $A_r = 150$nm and the layer distance $D = 1.846$µm). In the first case, when no scatterers are used, the angle range for which the far-field intensity is not negligible is estimated to be between $-6°$ and $6°$ (Fig. 3.3 a) and this is the range to which the number of pixels refers. In the second case, when four scatterer layers are present, the chosen angle range is between $-60°$ and $60°$ (Fig. 3.3 c). To be clear, in the case without scatterers, a smaller angle range with higher resolution is considered w.r.t. to the case with scatterers, so that the number of pixels is the same.

Let us stress that the expected value of the classification error and the confidence intervals drawn in the following plots are extracted from the results obtained from 20 random permutations of the simulated samples. The resulting error rates for different numbers of pixels and for different noise levels (Fig. 3.4) show that the use of scatterer layers allows for a significant error rate reduction (up to around 50%), provided that a sufficient number of pixels and a low enough noise level are considered. The increased sensitivity of classification performance towards added noise level when scatterers are used is ascribed to the fact that the scatterers' presence unfolds the cell diffraction pattern into a higher number of components (Fig. 3.3 c) that may be important for classification. Thus, it is probable that some of these components have low intensity with respect to the average pattern intensity and are therefore easily overcome by high relative levels of noise.

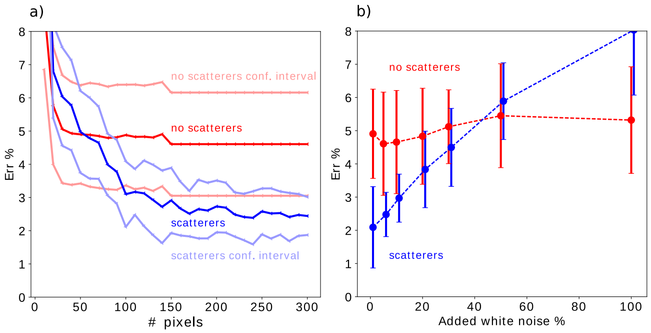


Figure 3.4: Comparison between the test error rates of "normal" and "cancer" cell classification, corresponding to the absence (in red) and the presence (in blue) of scatterers. A green laser source ($\lambda = 532$nm) is employed. **a)** Test error rate as a function of the number of employed pixels, with 5% added white noise. The darker and the lighter versions of the two line colors respectively represent the mean value and the confidence interval (of ± 2 standard deviations) over the 20 sample sets generated for validation. **b)** Test error rate (averaged over the values obtained considering $N_{\text{pix}} = 250, 260, ..., 300$) as a function of the percentage of added noise. In order to avoid error bar overlap, some of the blue points are slightly shifted to the right. Both the plots show that the scatterers' presence allows for an error rate reduction up to around 50%, provided that a sufficient number of pixels and a low enough noise level are considered.

3.4.2 Conceptual link between the physical system and a random neural network

In this subsection we take a closer look to the mathematics that allow us to consider the described physical system as an ELM network. The following simplified argumentation also provides us with useful insight on how to improve even further the computational capability of our machine learning approach. For example, here we discover that if there seems to be a limit to the improvements obtainable via the use of scatterers when a green coherent source is used, this limit can be overcome by decreasing the source wavelength.

Let us neglect for a moment the light deflection due to the cell's refractive index structure and let us consider only the phase shift of the light along all the possible fixed paths (here labeled with n) through the cell to one pixel on the screen. At the top of Fig. 3.5 a drawing representing three examples of these paths is shown. Let us state that the light along a path n has unitary initial amplitude and null initial phase (the reasoning is independent of the initial conditions) and that it accumulates a phase shift θ_n through the cell. Moreover, let us say that its amplitude is reduced by a factor A_n and its phase is increased by ϕ_n along the path to the pixel excluding the path inside the cell. Thus, the complex amplitude of the radiation impinging on the pixel is $\sum_n A_n e^{i(\theta_n+\phi_n)}$ and the acquired intensity I is proportional to:

$$I \propto \left|\sum_n A_n e^{i(\theta_n+\phi_n)}\right|^2 = C + \sum_{m<n} \left[A_{nm}\cos(\theta_n-\theta_m) + B_{nm}\sin(\theta_n-\theta_m)\right] \tag{3.5}$$

where C, A_{nm} and B_{nm} (that can also account for the presence of scatterers) are constants with respect to θ_n but depend on A_n and ϕ_n. These dependencies are omitted as the phase shifts θ_n are the only actual inputs of our classifying system, neglecting the light absorption in the cell. Eq. 3.5 shows that the phase-shift-to-intensity transfer function on the pixel can be written as a linear combination of all the possible sine and cosine functions whose argument is the phase shift difference between two of the considered optical paths (bottom of Fig. 3.5). Note that if the deflection of the light path due to the presence of the cell was also considered, we would have a richer dependence on θ_n in the right-hand side of Eq. 3.5 (C, A_{nm} and B_{nm} will also depend on θ_n). Nevertheless, the sines and the cosines in Eq. 3.5 would still be present and the following argument would still be relevant. It is important to note that, in this representation, the only role of the scatterer layers is to improve classification performance by providing more suitable weights A_{nm}, B_{nm} and C.

Let us now consider, for instance, the difference $\Delta\theta$ between phase shifts corresponding to a path through the nucleus and a neighboring path that instead does not intersect the nucleus. Let us call this phase shift difference $\Delta\theta_n$ in the case of a “normal" cell (smaller nucleus) and $\Delta\theta_c$ in the case of a “cancer" cell (bigger nucleus). We can intuitively say that if the readout linear classifier is able, for example, to detect the difference ΔI between the intensity contributions produced respectively by $\Delta\theta_n$ and $\Delta\theta_c$ among the other intensity contributions, the system can be successfully trained to carry out the classification task.

From Eq. (3.5) follows that an estimate of this critical intensity difference is given by:

$$\Delta I \propto A\left[\sin(\Delta\theta_c) - \sin(\Delta\theta_n)\right] + B\left[\cos(\Delta\theta_c) - \cos(\Delta\theta_n)\right] \tag{3.6}$$

$$\text{with} \quad \Delta\theta_c = \frac{2\pi D_c}{\lambda}(n_{\text{nucleus}} - n_{\text{cytoplasm}})$$

$$\text{and} \quad \Delta\theta_n = \frac{2\pi D_n}{\lambda}(n_{\text{nucleus}} - n_{\text{cytoplasm}})$$

Here A and B are constants, $D_c \approx 2.5$µm and $D_n \approx 1.2$µm are the average diameter of the “cancer” and the “normal” cell model respectively, λ is the wavelength of the considered light, $n_{\text{nucleus}} = 1.39$ and $n_{\text{cytoplasm}} = 1.37$ are the

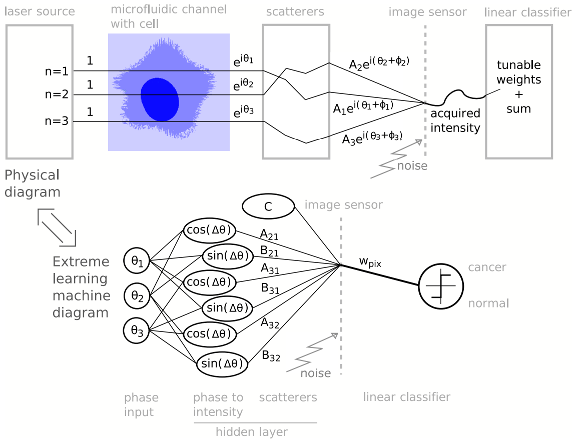


Figure 3.5: Two equivalent drawings describing the proposed classifying system. At the top, a physical diagram shows an example of amplitude and phase evolution along 3 optical paths that end up impinging on the same pixel of the image sensor. The acquired light intensity is then weighted and summed by a linear classifier. At the bottom, a diagram (under the form of a neural network architecture) represents the corresponding mathematical operations on the light phase accumulated through the cell refractive index structure (see Eq. 3.5).

refractive index of the nucleus and of the cytoplasm in the employed cell model. Let us stress that we expect a bad classification performance if the system has a too linear or a too random response, since generally ELM networks need nonlinear but not too chaotic internal node operation to perform well. It can be noted that in Eq. (3.5) these two undesired conditions may be ascribed to $\theta_n - \theta_m \ll \pi$ (linear regime) and $\theta_n - \theta_m \gg \pi$ (chaotic regime) respectively.

Let us know put some values to the terms in Eq. (3.6) when we focus on distinguishing different nucleus sizes. In particular, if $\lambda = 0.532$µm we have $\Delta\theta_c \approx 0.6$ and $\Delta\theta_n \approx 0.3$, which are quite smaller than π. By looking at the expressions for these two differences, it can be noticed that they can be increased by lowering the wavelength. This implies the need to employ an UV laser, which is usually significantly more expensive than its green counterpart and could damage the illuminated cells.

3.4.3 UV laser light

The reasoning discussed in the previous subsection suggested that the nonlinearity (and thus the computational power) of the considered ELM network can be increased by decreasing the wavelength of the employed laser light. Here we directly demonstrate this by adapting our FDTD simulations so that UV light is used to illuminate the simulated cell.

Let us first consider a system comprising 4 scatterer layers ($A_r = 150$nm and $D = 2.85$µm) with source wavelengths nm $= 532$nm, 400nm, 300nm and 200nm. The overall change in the acquired diffraction patterns was calculated for small modifications (of 130nm) of the nucleus size (Fig. 3.6), keeping the rest of the simulation parameters fixed. The change between two interference pattern arrays has been calculated by summing the absolute values of the elements of their element-wise difference array. The dispersion of the employed materials was accounted for and the same absolute contrast was kept between the refractive indexes in the cell model and the water.

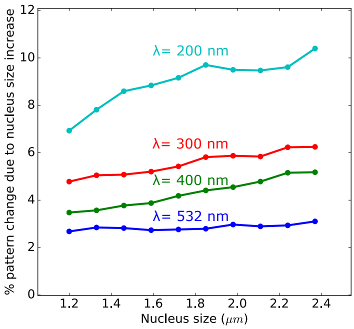


Figure 3.6: Change in the acquired diffraction pattern due to small increases (130 nm) of the nucleus size as a function of the starting nucleus size. The change between two interference patterns has been calculated by summing the absolute values of the elements of their point-wise difference vector. It can be noted that the smaller the employed wavelength is, the larger the pattern modifications becomes, implying an easier classification task.

As expected, the impact of nucleus modifications on the acquired interference pattern increases significantly as the laser wavelength is decreased. On the other hand, the same happens to the transmission losses due to the scatterers' presence (from 29% at $\lambda = 532$nm to 54% at $\lambda = 200$nm).

In order to consider a plausible UV laser source, the classification performance was investigated using $\lambda = 337.1$nm, which is the emission wavelength of nitrogen gas lasers. In particular, we compared (see Fig. 3.7) the case without scatterers and the case with 4 scatterer layers ($A_r = 150$nm and $D = 2.85$μm). In comparison with the results obtained using a green laser source (Fig. 3.4), the UV laser implementation substantially enhances the beneficial effects of the scatterers. In fact, not only is the achieved error rate reduction (up to an order of magnitude) remarkably larger, but also the robustness to noise is significantly improved. This confirms the predictions arisen from the discussion in the previous subsection: even a slight nonlinearity increase of the processing in the hidden nodes, due to a decrease in light wavelength, results in a considerable enhancement of the classification performance.

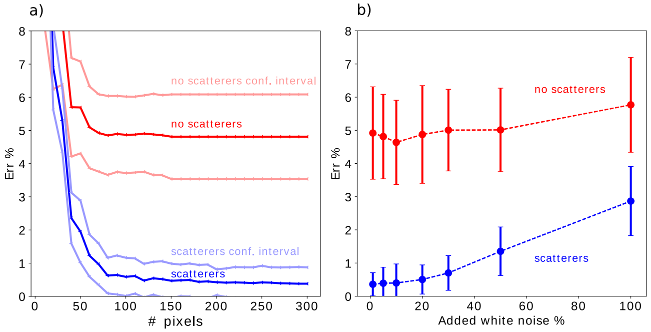


Figure 3.7: Comparison between the test error rates of "normal" and "cancer" cell classification, corresponding to the absence (in red) and the presence (in blue) of 4 layers of scatterers. An UV laser source ($\lambda = 337.1$nm) is employed. **a)** Test error rate as a function of the number of employed pixels, with 5% added white noise. The darker and the lighter versions of the two line colors respectively represent the mean value and the confidence interval (of ± 2 standard deviations) over the 20 sample sets generated for validation. **b)** Test error rate (averaged over the values obtained considering $N_{\text{pix}} = 250, 260, ..., 300$) as a function of the percentage of added noise. Both the plots show that the scatterers' presence allows for a considerable error rate reduction (up to an order of magnitude) in the entire investigated ranges of number of pixels and noise level.

However, when real applications are concerned, the option of employing UV lasers is highly impractical, as these light sources are usually quite expensive and would probably damage or even kill the illuminated cells. In the next subsection we discuss an alternative way to improve the classification performance by exploiting the same principle, while still using visible laser light.

3.4.4 Classification improvement by means of an optical cavity

In the previous subsection we considered the use of UV laser light, which implies an increase in the optical path through the given nuclei, and which in turn is expected to increase the nonlinearity of the cell information representation in the acquired interference patterns (see Subsection 3.4.2). In this subsection a more feasible solution is presented. It consists of increasing the effective optical path length through the cell by inserting it in an optical cavity. Intuitively, this makes the impinging light pass, on average, more than once through the cell.

In practice, in the FDTD simulation design, two Bragg reflectors are placed at the two external sides of the microfluidic channel, orthogonally to the light beam direction, creating a Fabry-Pérot cavity (Fig. 3.8). The employed Bragg reflectors are each composed of 3 layers of SiO_2 with a thickness of (455 ± 10)nm in a Si_3N_4 cladding. The error in the layer width was implemented by adding a random value sampled from an uniform probability distribution between -10 and 10nm. It approximately accounts for fabrication errors. The distance $D = 21.02$μm between the reflectors was chosen so that the portion of light passing through and near the nucleus of the cell was resonant. This was done by monitoring the light intensity inside the cavity for different values of D. Note that such a tuning was relatively easy to perform because the cell acts as a weak converging lens, providing an additional light confinement along the microfluidic channel direction that could be visually recognized.

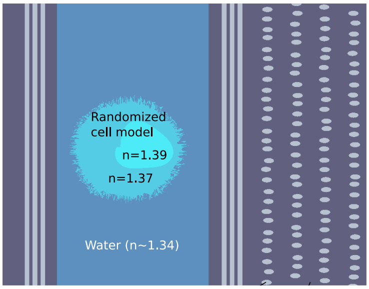


Figure 3.8: Sketch of the simulated area. The FDTD simulation is as described by Fig. 3.1, with the difference that an integrated Fabry-Pérot optical cavity composed by Bragg reflectors is placed next to the walls of the microfluidic channel containing the cell. The employed Bragg reflectors are each composed of 3 layers of SiO_2 with a thickness of (455 ± 10)nm in a Si_3N_4 cladding

The reflectivity R of the reflectors is also a crucial parameter, since it controls the cavity Q-factor and therefore controls both the sensitivity of the resonance to intracavity optical path lengths and how long the light stays, on average, inside the cavity. This means that, by tuning R, a trade-off has to be achieved between how much the phase shift due to the selected resonant cell part is increased and how much the corresponding resonance is stable. In this case, for instance, the sensitivity of the acquired intensity pattern to the nucleus size is to be improved by increasing the average time that the resonant light passing through the nucleus stays in the cavity. On the other hand, if the cavity Q-factor is too high, the resonance strength might be strongly influenced by uninteresting small details of the cell structure or by fabrication errors. Generally, the cavity should be designed so that the average light phase shift differences due to the optical feature of interest corresponding to the considered classes are roughly between $\pi/2$ and 2π, as we pointed out in Subsection 3.4.2. In particular, the reflectors employed in the simulations (composed of three layers) have a satisfying reflectivity of 56%, while it turned out that similar reflectors with 4 and 5 layers have a too high reflectivity, respectively of 73% and 85%. Additionally, in this case a longer simulation time was required with respect to the FDTD simulations without cavity, because the optical resonator needed time to fully charge.

Regarding the obtained classification results (see Fig. 3.9), the classification error for different numbers of pixels and for different noise levels shows a substantial improvement with respect to the green source case without cavity (Fig. 3.4). In particular, the classification improvement due to the use of scatterers is increased to a factor 5 by the cavity, for sufficiently low but still plausible noise levels ($< 10\%$). At these noise levels, the results are similar to what was obtained with an UV light source (Fig. 3.7), without the drawback of possible cell damage and overly expensive laser. For higher noise levels an increased sensitivity to noise pushes the classification error rate to significantly higher values.

As a final remark, an additional advantage arising from the use of an optical cavity is that it can potentially be designed to increase the intensity pattern sensitivity towards specific optical path lengths, making the optical features of interest more evident to the readout classifier with respect to other competing ones.

3.5 Further investigations of the classification technique

3.5.1 Generalization of the classification system: learning another task

It is worth wondering whether and how the proposed technique can generalize to other kinds of tasks. In fact, one of the key advantages of ELM implementations

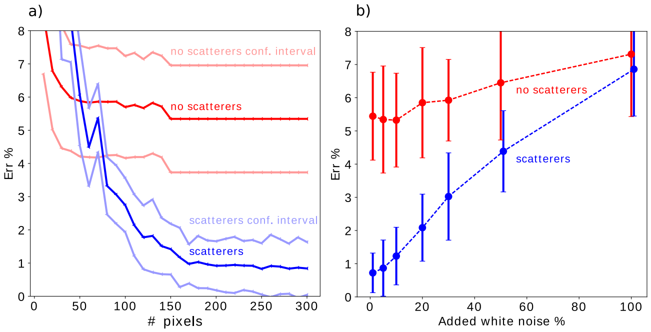


Figure 3.9: Comparison between the test error rates of "normal" and "cancer" cell classification, corresponding to the absence (in red) and the presence (in blue) of 4 layers of scatterers. A green laser source ($\lambda = 532$nm) was employed and a Fabry-Pérot cavity composed by integrated Bragg reflectors was placed at the sides of the microfluidic channel, as in Fig. 3.8. **a)** Test error rate as a function of the number of employed pixels, with 5% added white noise. The darker and the lighter versions of the two line colors respectively represent the mean value and the confidence interval (of ± 2 standard deviations) over the 20 sample sets generated for validation. **b)** Test error rate (averaged over the values obtained considering $N_{\text{pix}} = 250, 260, ..., 300$) as a function of the percentage of added noise. Both plots show that the combination of scatterers and optical cavity allows for a considerable error rate reduction (up to a factor 5) in most of the investigated ranges of number of pixels and noise level.

is that one network type can work for different applications [17]. In particular, it is important to show that different types of cell classification can be learned by the linear classifier, without the need for changing the hardware components of the classification system (e.g. the configuration of the dielectric scatters). Indeed, this would enable to employ the same cell cytometer for different applications, with the only requirement of retraining the readout weights applied to the pixel values of the acquired interference patterns.

In order to (non exhaustively) test this property, the classification of "lymphocyte" and "neutrophil" cells (see Subsection 3.2.1 and Fig. 3.2 b) was attempted using exactly the same hyperparameters of the previously discussed classification with a green light source, i.e. without performing any kind of further optimization other than the training of the linear readout classifier. We found that the classification improvement due to the use of scatterers (Fig. 3.10) is even greater than the one for "normal" and "cancer" cell classification (Fig. 3.4). Thus, the technique discussed in this chapter was proven to generalize to two fundamentally different classification tasks, as one is based on nuclei with different area but same shape, and the other on nuclei with same overall area but different shape.

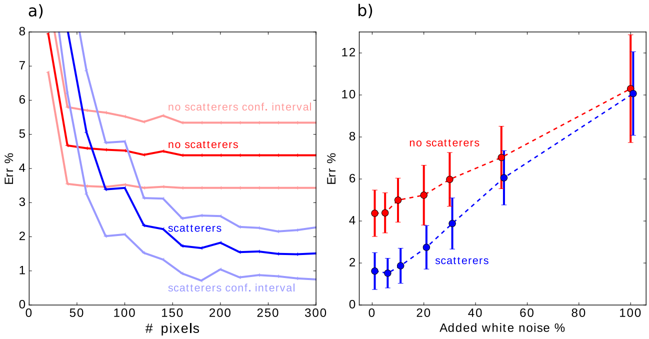


Figure 3.10: Comparison between the test error rates of "lymphocyte" and "neutrophil" cell classification, corresponding to the presence (in blue) and the absence (in red) of scatterers. The scatterer configuration and the light source are the same as for the results shown in Fig. 3.4. **a)** Test error as a function of the number of employed pixels, with 5% added white noise. The darker and the lighter versions of the two line colors respectively represent the mean value and the confidence interval (of ± 2 standard deviations) over the 20 sample sets generated for validation. **b)** Test error rate (averaged over the values obtained considering $N_{pix} = 250, 260, ..., 300)$ as a function of the percentage of added noise. In order to avoid error bar overlap, some of the blue points are slightly shifted to the right. Both plots show that the scatterers' presence allows for an error rate reduction greater than 50%, provided that a sufficient number of pixels and a low enough noise level are considered.

3.5.2 Cell classification using near-field interference patterns

All the results that have been presented so far are about cell classification employing interference patterns acquired by a virtual far field monitor, which can approximate be the case when the light intensity is measured by an image sensor placed relatively distant from the illuminated cell. Nevertheless, it can also be interesting to apply the classifier on patterns acquired with a near-field monitor, i.e. without the application of the far-field projection on the field calculated at the intensity monitor position (see Fig. 3.1). Indeed, in real life the image sensor has often to be placed as close as possible to the microfluidic channel in order to avoid the use of lenses and to build a compact device. In such a case, the measured field is expected to be somewhere between the near field and the far field.

Therefore it is important to check if the proposed classification system works for both these boundary cases. Moreover, the classification using the near field can provide an idea on whether the proposed classification system could work when, instead of a traditional image sensor, integrated photodetectors would be used to sample the output light. Furthermore, the near field could also be collected by waveguides and modulated before recombination, in order to perform the readout classifier operations in an all-optical way.

Before training and testing the readout linear classifier on the “normal” v.s. “cancer” cell classification, we plotted the median interference pattern and the corresponding 95% percentile range for the two cell classes and for the cases without and with 4 layers of scatterers (Fig. 3.11). We notice that the use of scatterers greatly modifies the projected near-field patterns, generating more abrupt and finer spatial variations. The effect of the cell class on the average patterns is slight but perceptible from the plots.

From the obtained classification results, we can conclude that the use of dielectric scatterers is relatively more beneficial (with a reduction in classification error of around a factor 4) when near-field patterns are considered as samples, compared to their far-field counterparts (Fig. 3.12). Indeed, even if a similar error is achieved when scatterers are used, the no-scatterers baseline error is significantly higher (almost double) in the near field case.

3.5.3 Classification performance for different scatterer configurations

Finally, in this subsection we present a small exploration of the classification performance when 9 different scatterer configurations are employed in the simulations (Fig. 3.13). In particular, we focused on the effect of the number of scatterer layers, of the random displacement amplitude (here called A_r) and of the dimension of the dielectric pillars.

In order to properly compare the classification performance of these different systems, it is important to take into account that one could be more prone to overfitting than another, and that therefore it may require more samples or fewer features (pixels) or stronger L2 regularization strength (see Chapter 1) in order to perform well. Thus, in this investigation we tried to remove the influence of these factors as much as we could, in order to achieve a fair comparison of the scatterer configurations for this classification task.

To do so, we employed 7200 simulated interference patterns for each scatterer configuration (more than double w.r.t. the previously presented classifications). Moreover, the classification errors were obtained through two nested 5-fold cross-validation cycles: the inner one was used to optimize both the L2 regularization strength (chosen among $10, 25, 50, 100, 500$) and the image sen-

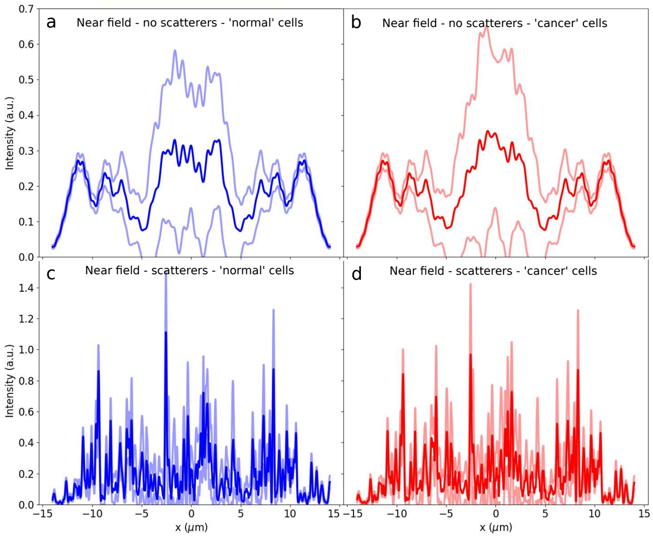


Figure 3.11: Comparison of the median near field interference patterns projected by simulated "normal" cells (red) and "cancer" cells (blue). The dark lines represent the median of all the calculated patterns for a given cell class and scatterer configuration. The lighter lines represent the confidence interval, given by the 95% percentile range. **a**, **b**: patterns obtained without including dielectric scatterers in the simulations. **c**, **d**: patterns obtained considering 4 layers of dielectric scatterers (as in Fig. 3.1). The use of scatterers greatly modifies the projected near field patterns, generating more abrupt and finer spatial variations. The effect of the cell class on the average patterns is slight but perceptible from the plots.

sor resolution (the number of pixels is chosen among 300, 400, 500, 600, 700); the outer one to provide error bars to the performance estimations, i.e. to estimate the variance of the classification accuracy (see Chapter 1). Given the large number of simulations required, we chose to limit our exploration to the 9 configurations discussed here.

The classification results show that all the explored scatterer configurations provide a significant improvement w.r.t. the case without scatterers, both when near-field and far-field samples are considered (Fig. 3.13). Furthermore, taking into account the variance of the error estimations, we can notice that the performance improvement is similar for all the considered configurations, with only relatively small fluctuations. This suggests that the observed classification im-

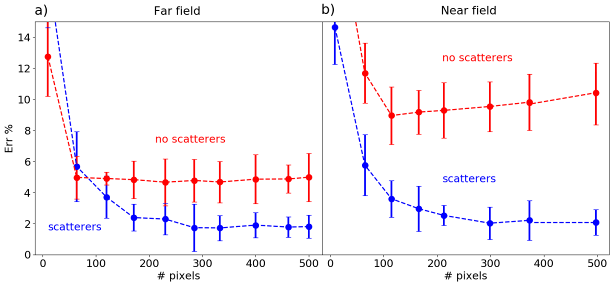


Figure 3.12: Comparison between the test error rates of "normal" and "cancer" cell classification, corresponding to the absence (in red) and the presence (in blue) of 4 layers of scatterers. Test error as a function of the number of employed pixels, with 1% added white noise. The the dots and the error bars respectively represent the mean value and the confidence interval (of ± 2 standard deviations) over the 20 sample sets generated for validation. **a)**, **b)** Respectively far-field and near-field interference patterns were considered as samples for the training and test of the readout linear classifier. In the two cases a similar error is obtained when scatterers are used, while significantly higher error is obtained in the near-field case compared the far-field case, when no scatterers are used.

provement can be obtained for a wide choice of scatterer configurations. In a practical implementation, this also shows that the proposed classification technique is robust against fabrication errors or accidental modifications that may affect the optical stage.

3.5.4 Results interpretation

The main contribution of this chapter is to numerically demonstrate that by means of interposed dielectric scatterers, whose size is comparable to the laser wavelength, it is possible to substantially enhance the *linear separability* (see Chapter 1) of acquired intensity patterns. This is a rather theoretical proof-of-principle that can be generalized to other optical machine learning implementations, even though we consider the specific case of label-free flow cytometry. Here we shortly discuss about the causes behind the observed increase in linear separability. Moreover, we point out the differences between the proposed optical approach and the application of a random nonlinear transformation (according to the ELM paradigm) after image acquisition.

First, let us better clarify what we mean when we state that the classification performance is enhanced because of an increase in *dimensionality* of the cell information representation. In fact, the most straightforward way to increase the

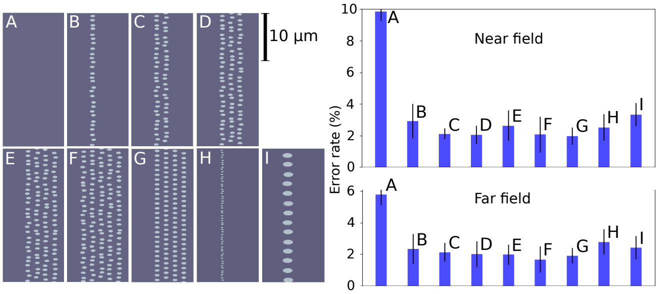


Figure 3.13: Test error rates (blue bars, on the right) of "normal" and "cancer" cell classification for different scatterer configurations (on the left) employed in the simulations. The capital letters show the configuration-error correspondence. The upper and the lower bar plots respectively show the results concerning the near-field and the far-field pattern classification. The classification errors were obtained through two nested 5-fold cross-validation cycles: the inner one was used to optimize both the L2 regularization strength and the image sensor resolution, the outer one to provide error bars to the performance estimations (vertical black segments on top of the blue bars, representing a confidence interval of ± 2 standard deviations). For each scatterer configuration, 7200 simulation results were employed as samples, more than double w.r.t. the previously presented classifications.

dimensionality of the obtained samples would be to increase the number of pixels of the image sensor. However, if the values measured at these additional pixels are strongly correlated (or linealry dependent) with the values measured at the other pixels, we say that the added features are *redundant* and they are not helpful in increasing the linear separability of the samples. Indeed, even if more redundant features are employed, the samples populate only a subspace of the feature space, with no effective increase in the dimension of their representation. Thus, adding redundant features does not improve the classification performance, no matter their number. Moreover, in order to add linearly independent features, these must be generated by a nonlinear transformation, hence the importance of nonlinearity in sample representation.

However, even adding non-redundant features does not necessarily improve the classification, since they might contain information that is not helpful in distinguishing different classes of samples, i.e. they might not be *relevant*. For example, values at noisy pixels are likely to be uncorrelated with other pixel values, but they clearly cannot be employed for the classification purpose. Indeed, the dimensionality of the class-specific information cannot be enhanced by adding irrelevant features. On the contrary, noise can overshadow relevant information and reduce class separability.

Therefore, in this context, by dimensionality enhancement we mean that the overall number of non-redundant *and* relevant features (or pixels) has increased. Or, more generally, that the overall non-redundancy and relevance has increased in the sample representation.

Intuitively, as it is suggested by Figs. 3.3 and 3.11, the use of scatterers allows to spread the particle information in a more diverse way on the image sensor (lower redundancy), e.g. by reducing the spatial correlation between neighbouring pixels. Moreover, it can increase the likelihood that class-specific information is not covered by irrelevant information (higher relevance). These statements were confirmed at a later time through suitable statistical analysis (see Chapter 4, Fig. 4.18), which showed the correlation between the classification performances reported in Fig. 3.13 and the employed measures for redundancy and relevance of the pixel values. In the light of these considerations, it is not so surprising that similar classification performances are obtained employing different scatterer configurations (as shown in Fig. 3.13). Indeed, the requirements for classification improvement are relatively general and they can be achieved in many different ways, in line with the ELM paradigm.

Finally, it is interesting to note that an alternative ELM approach could be implemented by applying randomly chosen nonlinear transformations to each pixel values, in the electric domain after image acquisition, and feeding the transformed values to a linear classifiers. By employing suitable analog electronic hardware, it would be possible to perform this transformation at high enough speed and low latency. However, such an implementation presents fundamental disadvantages w.r.t. leveraging the optical domain as in the proposed approach. To begin with, additional energy would be required if active electronic components are employed for the nonlinear transformation. More interestingly, the electronic ELM implementation could increase the linear separability of the samples only by decreasing feature redundancy, but it could not improve the relevance of the pixels. Indeed, if on one pixel the class-related information is overshadowed by noise (e.g. by an optical signal that does not correlate with class labels), there is nothing that a nonlinear transformation in the electric domain could do to overcome this issue. On the other hand, by acting in the optical domain, the full 3D light propagation can be modified and leveraged to improve pixel relevance. Moreover, the effect of dielectric scatterers on the acquired interference patterns benefits from simple and broad reconfigurability, e.g. the position of the scatterers w.r.t. the flowing cell or the camera can be easily changed and optimized.

It should be stressed that the design and the realization of the experiment presented in the next chapters are based on the results interpretation here discussed.

3.6 Summary and conclusion

In this chapter we presented a proof of concept, based on FDTD simulations, of a new hardware-based machine-learning technique for biological cell classification in label-free flow cytometry. In particular, we proposed an ELM approach based on the simple interposition of microscopic dielectric scatterers between the illuminated cell and the image sensor. This is meant to enhance the classification power of a linear classifier applied to the pixel values of acquired interference patterns projected by cells. The scatterers allow to enrich, by random optical mixing, the nonlinear relation between the refractive index structure of the cell and the intensity pattern acquired by the image sensor. This operation occurs at the speed of light (without considering the image acquisition time), and therefore the only required computational cost of the classification process is given by a linear classifier in the software domain. Thus, the proposed technique aims to speed up the operation of imaging flow cytometers, whose throughput is often limited by the high computational cost of classification algorithms. Moreover, the training of a linear classifier is significantly easier and computationally cheaper w.r.t. more complex machine learning-models, such as convolutional or deep NNs.

The main classification task we considered for our proof-of-concept was to distinguish the far-field interference patterns projected by cells with two different average nucleus sizes. In each simulation, the employed randomized cell model generated a cell with different shape and position, which is a relatively realistic source of noise from the perspective of the task to be learned. Moreover, white noise was added to the pixel values of the simulated interference patterns. For each different optical configuration we considered, thousands of FDTD simulations were run in order to generate enough samples (i.e. simulated interference patterns) to train and test the readout linear classifier. In each case, we evaluate the classification performance improvement (w.r.t. similar cases without dielectric scatterers) associated to the optical configuration under investigation. Put differently, we checked if and how the dielectric scatterers could modify the simulated interference patterns so that they were more easily classified by the linear classifier applied to the pixel values. This evaluation was generally performed for different values of added white noise and for different numbers of pixels (i.e. resolutions) in the simulated intensity monitor.

We first showed that 4 layers of dielectric scatterers could improve the classification accuracy when a green laser was considered as a light source. Then we discussed the mathematical link between an ELM neural network and the proposed classification system. The obtained insight suggested that the network nonlinearity (and therefore its potential computational power) could be enhanced by increasing the optical path of the light sent through the cell. A way to do so, is to lower the wavelength of the simulated light source. Indeed, we showed that a significantly greater improvement could be achieved by considering UV light. Another more feasible way to increase the optical path is to enclose the cell in an optical cavity. We considered a green laser again and a Fabry-Pérot cavity made by integrated Bragg reflectors placed at the outer sides of the microfluidic channel walls. As expected, the optical cavity could enhance the classification improvement due to the presence of the scatterer.

In order to show that the same optical configuration could be beneficial in learning another machine learning task, we considered the classification of cells based on the nucleus shape (instead of on the nucleus size), by suitably modifying the randomized cell model. We found that the 4 layers of scatterers could provide an even larger improvement in the classification accuracy for this new task.

Then, we investigated how the classification technique performs when near-field interference patterns are considered as samples, as opposed to the far-field patterns used before. In relative terms, the use of scatterers is in this case more beneficial w.r.t. the far-field case. Indeed, the error when no scatterers are used is almost double, while with scatterers a similar accuracy is obtained as in the far field.

Finally, we presented the outcome of a small exploration of the performance provided by the use of 9 different scatterer configurations. In this case, more samples and a more complicated validation algorithm were required to ensure that the obtained classification error estimations could be compared in a fair way. Surprisingly, we found that all the tested configurations provided a similar improvement w.r.t. the case without scatterers, both in the far-field and in the near-field cases.

In conclusion, the presented numerical investigation suggests that employing simple dielectric scatterers in a label-free imaging flow cytometer could significantly simplify the learning of classification tasks, when a fast and easy to train linear classifier is employed. Such a classification improvement was observed in a relatively wide variety of cases, so that the proposed technique seems suitable for robust and cheap implementations in real-life devices. Moreover, our approach can be easily applied to extreme high-throughput label-free flow cytometers based on optofluidic time-stretch microscopy [18, 19] (see Chapter 0). Indeed, these cytometers produce cell images at such a high rate (> 10000 particles/s) that online operation is often not possible due to the too high computational cost of the processing algorithms.

In the next chapter, we present an attempt to experimentally demonstrate the presented technique employing a simple flow cytometer for the classification of microparticles based on their dimensions.

References

[1] Yuqian Li, Bruno Cornelis, Alexandra Dusa, Geert Vanmeerbeeck, Dries Vercruysse, Erik Sohn, Kamil Blaszkiewicz, Dimiter Prodanov, Peter Schelkens, and Liesbet Lagae. *Accurate label-free 3-part leukocyte recognition with single cell lens-free imaging flow cytometry.* Computers in biology and medicine, 96:147–156, 2018.

[2] Bendix Schneider, Joni Dambre, and Peter Bienstman. *Fast particle characterization using digital holography and neural networks.* Applied optics, 55(1):133–139, 2016.

[3] Young Jin Heo, Donghyeon Lee, Junsu Kang, Keondo Lee, and Wan Kyun Chung. *Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip.* Scientific reports, 7(1):1–9, 2017.

[4] Bruno Cornelis, David Blinder, Bart Jansen, Liesbet Lagae, and Peter Schelkens. *Fast and robust Fourier domain-based classification for on-chip lens-free flow cytometry.* Optics Express, 26(11):14329–14339, 2018.

[5] Yueqin Li, Ata Mahjoubfar, Claire Lifan Chen, Kayvan Reza Niazi, Li Pei, and Bahram Jalali. *Deep cytometry: deep learning with real-time inference in cell sorting and flow cytometry.* Scientific reports, 9(1):1–12, 2019.

[6] Alessio Lugnan, Emmanuel Gooskens, Jeremy Vatin, Joni Dambre, and Peter Bienstman. *Machine learning issues and opportunities in ultrafast particle classification for label-free microflow cytometry.* Scientific Reports, 10(1), 2020.

[7] Allen Taflove and Susan C Hagness. *Computational electrodynamics: the finite-difference time-domain method.* Artech house, 2005.

[8] Alessio Lugnan, Joni Dambre, and Peter Bienstman. *Integrated pillar scatterers for speeding up classification of cell holograms.* Optics express, 25(24):30526–30538, 2017.

[9] Andrew Katumba, Matthias Freiberger, Floris Laporte, Alessio Lugnan, Stijn Sackesyn, Chonghuai Ma, Joni Dambre, and Peter Bienstman. *Neuromorphic computing based on silicon photonics and reservoir computing.* IEEE Journal of Selected Topics in Quantum Electronics, 24(6):1–10, 2018.

[10] Kristof Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, Geert Morthier, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman. *Experimental demonstration of reservoir computing on a silicon photonics chip.* Nature communications, 5(1):1–6, 2014.

[11] K-W Wong, C-S Leung, and S-J Chang. *Use of periodic and monotonic activation functions in multilayer feedforward neural networks trained by extended Kalman filter algorithm.* IEE Proceedings-Vision, Image and Signal Processing, 149(4):217–224, 2002.

[12] Liesbet Lagae, Dries Vercruysse, Alexandra Dusa, Chengxun Liu, Koen de Wijs, Richard Stahl, Geert Vanmeerbeeck, Bivragh Majeed, Yi Li, and Peter Peumans. *High throughput cell sorter based on lensfree imaging of cells.* In 2015 IEEE International Electron Devices Meeting (IEDM), pages 13–3. IEEE, 2015.

[13] Daniele Zink, Andrew H Fischer, and Jeffrey A Nickerson. *Nuclear structure in cancer cells.* Nature reviews cancer, 4(9):677–687, 2004.

[14] Shirley Mitchell Lewis, Barbara J Bain, Imelda Bates, and John Vivian Dacie. *Dacie and Lewis Practical Haematology.* Churchill Livingstone, 2011.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[16] Chris M Bishop. *Training with noise is equivalent to Tikhonov regularization.* Neural computation, 7(1):108–116, 1995.

[17] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. *Trends in extreme learning machines: A review.* Neural Networks, 61:32–48, 2015.

[18] K. Goda, K. K. Tsia, and B. Jalali. *Serial time-encoded amplified imaging for real-time observation of fast dynamic phenomena.* Nature, 458(7242):1145–1149, apr 2009.

[19] Cheng Lei, Hirofumi Kobayashi, Yi Wu, Ming Li, Akihiro Isozaki, Atsushi Yasumoto, Hideharu Mikami, Takuro Ito, Nao Nitta, Takeaki Sugimura, Makoto Yamada, Yutaka Yatomi, Dino Di Carlo, Yasuyuki Ozeki, and Keisuke Goda. *High-throughput imaging flow cytometry by optofluidic time-stretch microscopy.* Nat. Protoc., 13(7), jul 2018.

4

Development of proof-of-concept microsphere classification based on optical extreme learning machine

In this chapter we present the development of an experiment regarding the classification of transparent PMMA microspheres (called also microbeads or microparticles) on the basis of their diameter. As in the previous chapters, we consider an imaging and label-free microflow cytometry application, where a lensless digital holographic microscopy setting is employed, without reconstructing the particle images from the acquired holograms.

The main aim of the experiment was to demonstrate that very fast particle classification based on machine learning is possible by directly applying a linear classifier (or readout classifier) on the pixels of the recorded images, exploiting the extreme learning machine paradigm (see Chapter 1). We also investigated whether the classification performance can be enhanced by interposing scatterer layers between flowing microparticles and the image sensor, trying to validate the results obtained through FDTD simulations presented in Chapter 3. Overall, we chose not to analyse and treat the noise in the acquired images as it is conventionally done in microscopy, e.g. by minimizing spurious Fabry-Pérot fringes with anti-reflection coatings or by employing a beam expander to reduce the effects of the diffraction at the pinhole. Instead, since we did not aim for any image reconstruction, we directly focused on adjusting the experimental parameters in order to improve the classification performance. In general, our approach was to try to keep the setup a simple as possible, and to learn step by step the requirements for a successful application of a linear classifier.

It should be stressed that the experiment was designed to investigate and overcome the issues (mainly regarding measurement bias) encountered in the work presented in Chapter 2. To do so, we had to build a setup from scratch and learn the best practices to achieve a high enough quality of measurements. Because of these reasons, we had to consider an experiment that is representative but also simple and versatile, since the possibility of performing many test measurements was a key requirement, as well as the option of changing the setup and the measurement configuration accordingly. Therefore, we could not employ white blood cells as in Chapter 2, since it would have dramatically increased the cost and complexity of measurements. Taking these requirements into consideration, together with the insight gained through the numerical investigations discussed in Chapter 3, we decided that the classification of transparent PMMA microbeads of different sizes was a suitable proof-of-principle application. The natural continuation of our experiment would be to apply the learned methodology to the classification of biological cells, which is more interesting both from a scientific and an industrial perspective.

This chapter summarizes the most meaningful steps of the experiment development, from the construction and test of the first version of the employed setup and measurements, to the final version, which eventually provided samples suitable for a reliable particle classification with satisfying accuracy. The first advances in development were mainly due to the work of Jeremy Vatin and Emmanuel Gooskens, respectively in the context of his internship and of his Master's Degree Dissertation, and summarized in the first two Sections respectively. Let us now provide a brief outline of the different versions of the employed setups, each corresponding with the first four sections of this chapter. In Section 4.1 we consider the first version of the setup, where the particle interference patterns modified by a phase *spatial light modulator* (SLM, employed as a scatterer layer) and the unmodified patterns were recorded at the same time with two different image sensors. In Section 4.2 we focus on the microfluidic system and hologram generation (momentarily removing the SLM from the setup) to analyse and overcome various issues encountered in the previous particle classification attempt. In Section 4.3 we resumed the use of the SLM, but the setup was modified so that both the modified and the unmodified particle patterns were recorded by the same camera. In Section 4.4, instead of the SLM, that turned out to introduce too much noise in the acquired images, we employed fixed physical scattering media of different kind and in various configurations.

It should be stressed that, together with the experimental work summarized in this chapter, a significant amount of time and effort was spent by Jeremy Vatin and Emmanuel Gooskens in trying to predict the outcome of the free-space experiment by means of simulations. In particular, suitable Fresnel diffraction equations were numerically solved. However, the simulation of the experiment turned out to be particularly complicated, e.g. because of the high computational power and memory required to calculate detailed diffraction due to a microfluidic channel, a microparticle and a scatterer layer. In particular, the mix of microscopic and macroscopic dimensions and the distances between the components was problematic (midway between the near field and the far field). Therefore, only a qualitative agreement with the experiment could be achieved, and these simulations did not provide a significant contribution to the experiment development. Thus we decided not to present the simulations in this chapter.

In the next chapter, the final version of the experiment and the corresponding results are discussed in details. Therefore, here we do not discuss the employed setups, measurements and analysis in detail, but we aim to describe to the reader the chain of observations, reasoning and choices that brought us to the final realization of the experiment. At the end of this chapter (in Section 4.5), we list the most important practical aspects that were learned during the development of the experiment and that turned out to be critical for its correct realization.

4.1 Initial two-branched setup

Most of the work presented in this section was carried out by Jeremy Vatin in the context of his internship at the Photonics Research Group.

4.1.1 Particle signal generation

The light beam generated by a helium-neon red laser (with wavelength of 632.8 nm and 3.5 mW emission power) was focused by a lens on a pinhole (with diameter of 10 or 25 μm) through which it illuminated a straight microfluidic channel integrated in a PMMA (plexiglass) slide.

The microfluidic channel had a cross section of 50 μm × 50 μm and its input and output were connected by silicone tubes to a beaker and a syringe respectively (see Fig. 4.1 and 4.2). The beaker contained a mixture of water, transparent PMMA microparticles (to be classified on the basis of their diameter, as an example see Fig. 4.3) and a small amount of surfactant (Triton X-100, to prevent particles from clustering or from attaching to walls). During measurements, the mixture in the beaker was constantly agitated by a magnetic stirrer, to avoid the deposition of the microparticles on the bottom. Indeed, it was very important to control the amount of particles entering the microfluidic channel: too few flowing particles resulted in too few pictured particle patterns, while too many flowing particles could easily clog the channel, causing the interruption of the measurement.

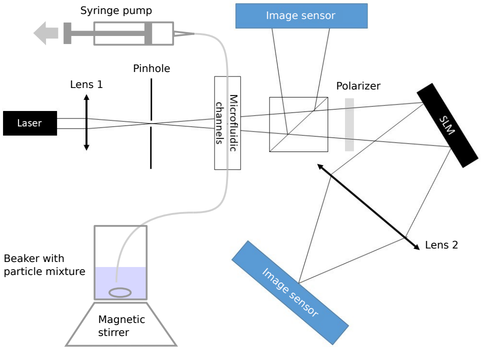


Figure 4.1: Drawing of the two-branched setup initially employed (a picture is shown in Fig. 4.2). A red laser light is focused by a lens into a pinhole. The transmitted beam illuminates a transparent microfluidic channel through which microparticles were made to flow. The forward scattered light is split into two equal beams: one is directly acquired by a camera (Thorlabs DCC1545M CMOS, 1280 x 1024 resolution), while the other is made to pass through a polarizer before reflecting on an SLM (Meadowlark Optics XY Phase Series, 512 x 512 resolution). The reflected beam is then focused by a lens on a second camera (Ximea MQ013MG-ON, 1280 x 1024 resolution).

The syringe at the other end of the particles path was actuated by a syringe pump, so that it sucked the mixture from the beaker through the microfluidic channel. The option of pulling the mixture with the syringe pump, as opposed to pushing it, was initially chosen because the magnetic stirrer could not fit inside a syringe. However, as we will see in Section 4.2, subsequently we had to switch to a pushing operation, since we discovered that pulling caused the generation of small bubbles at the interface between the input silicone tube and the microfluidic channel. Such bubbles provided undermining noise in the measured patterns and they could easily be mistaken for microparticles.

The particle mixture collected in the syringe was initially reused few times before being discarded. Also this practice was abandoned later on, since dust and dirt particles could easily accumulate providing unwanted perturbations in the acquired patterns.

4.1.2 Two output branches

The first employed setup was characterized by the use of two output branches and two cameras (see drawing in Fig. 4.1 and picture in Fig. 4.2): one to directly acquire the forward scattered light coming from the illuminated particles (here

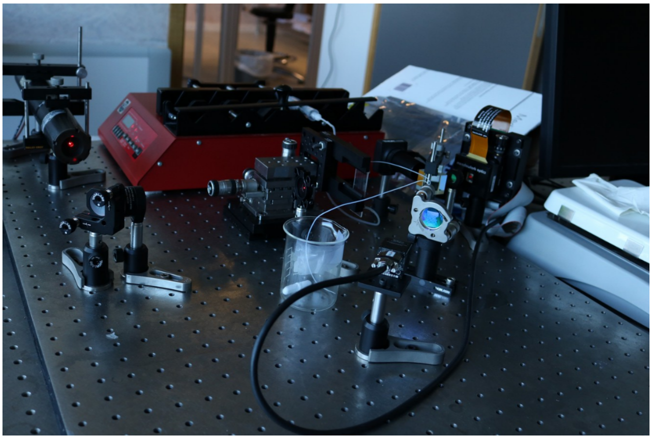


Figure 4.2: Picture of the two-branched setup initially employed (a corresponding drawing is shown in Fig. 4.1).

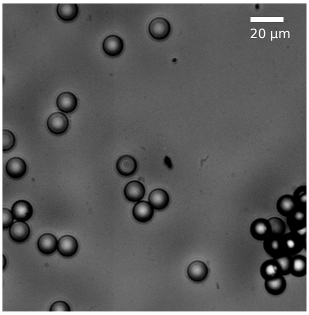


Figure 4.3: Microscope image of the mixture containing transparent PMMA microparticles, from the Master's Degree Dissertation of Emmanuel Gooskens [1].

also referred to as the *particle signal*) , the other to acquire the particle signal after it was further scattered by a phase *spatial light modulator* (SLM), which allows to apply an arbitrary phase shift spatial pattern to a light beam. The SLM was meant to play the role of *scatterer layer*, with reference to Chapter 3.

The first output branch was employed to acquire *direct interference patterns* (i.e. not modified by the scatterer layer) in order to unmistakenly detect when a particle was passing through the illuminated area. Indeed, detecting particle patterns was not as trivial as it may seem, since they could be easily confused with fluctuations in the measurement conditions, or the presence of bubbles or dirt in the particle mixture. As we will see later on in Section 4.2, the correct detection of particles (regardless of their class, as opposed to particle classification), which is here referred to as *background detection* or *labeling* to avoid confusion, required a rather complicated operation and a relatively large amount of work to be perfectioned. The images acquired with this output branch could also be used as samples for the particle classification without scatterers, so that further measurements without SLM could be avoided. The obtained results were used as a baseline to evaluate the performance improvement due to the SLM scattering action.

The second output branch was used to acquire *SLM patterns* (i.e. modified by a scatterer layer implemented by the SLM) to train and test a readout linear classifier. The scatterer layer, as in the previous chapter, was meant to control and enrich the internal connections of a corresponding hardware-based *extreme learning machine* (ELM), that is to improve the classification power of the readout classifier.

The employed SLM (Meadowlark Optics XY Phase Series, 512 x 512 resolution) is able to reflect vertically polarized light with an arbitrary phase shift at each pixel. Since it was built to work with infrared light, particular attention had to be paid in order to translate the nominal programmed phase shifts into the actual phase shifts exerted on the reflected red laser light. Such a device was meant to play a role analog to the dielectric scatterers investigated in the previous chapter through optical simulations. Its programmability allowed to easily explore the use of different scatterer configurations.

In order to make sure that the two cameras would acquire patterns at the same time, they had to be properly configured so that they could operate in a synchronized way. This turned out to be more complicated than expected, since it could not be done using the Matlab interface, but coding using the .NET framework had been necessary. Moreover, the requirement of having the two cameras synchronized, prevented them to operate in free run mode, and therefore limited their frame rate. In addition, the maximum frame rate was determined by the slower Thorlab camera. These limitations to the acquisition speed resulted in a much longer measurement duration and a larger amount of particles used, compared to the case where only the fastest Ximea camera was employed.

4.1.3 Measurement bias detected in the classification results

The described setup was employed to try to classify transparent PMMA microparticles of two different diameters: 13µm and 15µm. This was done similarly as in the previous chapter, that is by applying a machine learning linear classifier (logistic regression) to the pixel values of the acquired interference patterns. In particular, the main goal was to demonstrate an improvement in classification accuracy due to the optical mixing provided by the SLM.

The particle interference patterns belonging to the two particle classes were acquired in two separate measurements. Moreover, a third class was considered in the classification, which consisted of the *background images* acquired in both measurements. As in Chapter 2, we call background images the acquired patterns that are not significantly perturbed by the presence of a particle in the illumination area. In particular, to automatically detect background images, the overall intensity of each acquired image was computed and it was subtracted from the overall intensity of the first acquired image (which was considered a reference background image). The absolute value of this difference (i.e. an intensity fluctuation) was then computed for each image and compared to a chosen threshold value. The images whose intensity fluctuation was lower than the threshold were labelled as backgrounds. In Section 4.2 we will see that the background detection algorithm was subsequently revised and improved.

The classification results obtained in this first attempt were heavily affected by *measurement bias* (see Chapter 2), a specific type of *shortcut learning* [2], defined as the bias in the training of a machine learning model due to the correlation between the measurement noise in the training samples and the corresponding labels.

Let us now summarize the obtained results and then explain how they are interpreted.

First, the linear classifier was trained and tested without using the SLM. The test sample set consisted of 11,686 background patterns, 1807 patterns of 13µm particles and 2516 patterns of 15µm particles. For the sake of brevity, let us call these three classes respectively *class 0, A and B*. Overall, a relatively high classification accuracy was obtained for all classes, i.e. 88 %, 95% and 98% respectively, with a total accuracy of 90.2%. These results, that at first sight might seem promising, are suspicious in that the classifier seemed to learn more easily to distinguish one particle class from the other, rather than distinguishing a particle pattern from a background pattern. Indeed, this is unexpected, because the difference between particle classes has a much smaller influence on the acquired patterns than the difference between the absence and the presence of particle signal. Therefore, this difference is unlikely to be more exploitable for classification purposes.

This issue became much more evident when the SLM was used, i.e. when the light carrying the particle information was further mixed by a programmed phase profile. Without going too much into detail (since we are still discussing one of the many transient versions of the setup and of the classification process), the same classification process was tested for two different random phase profiles actuated by the SLM (see Fig. 4.4). In both cases, particle patterns were classified with high accuracy, while most of the background patterns were wrongly classified as particles. In particular, the background patterns recorded during the measurement of class *A* particles and during the measurement of class *B* particles were respectively classified as class *A* and class *B*. This clearly shows that the classifier could easily learn how to distinguish the patterns from slight differences in measurement conditions that characterized the two recording sessions. In other words, the training of the classifier was affected by measurement bias. It should be noted that if the background were not treated as an additional class, it would have been difficult to infer the presence of shortcut learning from the classification results, and the classifier would have seemed to have successfully learned the classification task.

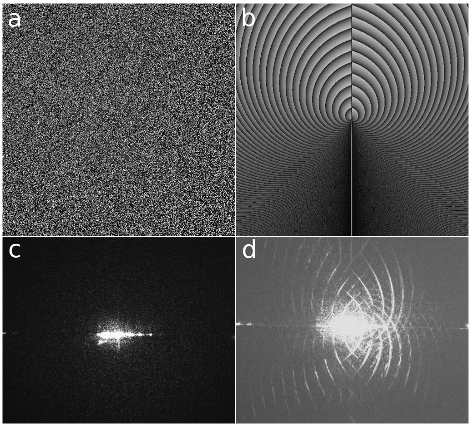


Figure 4.4: *a*, *b*: programmed phase shift patterns on the SLM, respectively projecting the patterns in *c* and *d*. The images were originally presented in the internship report of Jeremy Vatin.

On the other hand, when the SLM was used, the classifier could not detect the perturbation of the patterns caused by the presence of the particle, as shown by the fact that the background samples could not be correctly classified. This suggests that the *signal-to-noise ratio* (SNR) was not sufficiently high. In this case, when referring to the SNR, it should be understood that the noise is given by the background pattern and all the pattern perturbations that are not caused by the particle presence, while the signal is the representation of the particle information in the acquired intensity patterns. A separate classification attempt, considering as classes the background and particle patterns from the same measurement, confirmed that the classifier could not learn to detect the particle signal.

Another issue that is likely to have affected this setup, is the generation of bubbles in the fluidic circuit, probably at the interface between the input tube and the microfluidic channel. This is suggested by the fact that the automatic background detection algorithm, that was applied prior to the classification algorithm to label the background patterns, detected the presence of particles in groups of consecutive frames. From our subsequently gained experience, the particle patterns usually appear in single and separated frames, due to the high speed of the fluid in the microfluidic channel. Bubbles, on the other hand, generate stronger pattern perturbations in many consecutive frames.

In conclusion, this first setup and the corresponding measurements and classification pipeline were affected by at least three undermining issues:

- The measurements were conveying biasing information to the training algorithm of the classifier, causing measurement bias.
- The SNR in the acquired samples was not high enough, especially when optical mixing was actuated by the SLM.
- Bubbles were likely generated in the fluidic circuit, and were mistaken for particles.

In Section 4.2 we describe how we addressed these problems, among others.

4.2 Further measurement analysis and improvements

Most of the work presented in this Section was carried out by Emmanuel Gooskens in the context of his Master's Degree Dissertation [1]. In order to acquire more control over the setup and properly address the issues described in the previous Section, a step back was taken and a setup configuration without SLM was considered (see Fig. 4.5). The employment of the SLM was resumed afterwards (see Section 4.3).

The beam splitter was temporarily removed and the Ximea camera was placed directly in front of the microfluidic channel. The use of the fast camera alone allowed to considerably increase the acquisition frame rate to 170 fps, with a exposure time of 30µs. Such a short exposure time prevented motion blur from affecting the recorded particle signals.

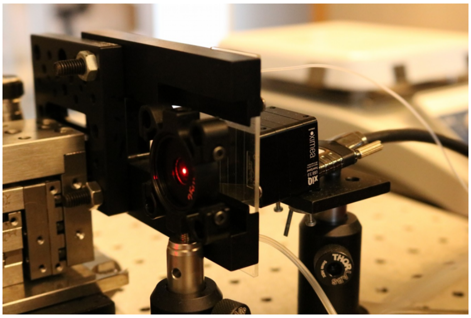


Figure 4.5: Detail of the setup without SLM, used to address the issues encountered using the initial two-branches setup. From left to right: the laser beam is focused on a pinhole, which is fixed near a slide containing straight microfluidic channels. The laser light passing through the pinhole and the channel is measured by an image sensor (Ximea CMOS camera).

4.2.1 Particle mixture improvement

In this application, it was essential to work with well-prepared and well-kept particle mixtures. Indeed, the particle density had to be carefully chosen in order to avoid clogging of the microfluidic channel, while making sure, at the same time, to record enough particle samples to train and test the classifier. Because of different factors, such as the increasing probability of clogging with time, the measurement sessions could not exceed few minutes.

Moreover, it was important to avoid formation of particles clusters or attachment of particles to the walls of the fluidic circuit, as these phenomena could favor channel clogging and the acquisition of unwanted patterns projected by more than one flowing particle. In order to lower the probability of these issues, a suitable amount of surfactant was dissolved in the particle mixtures. Furthermore, formation of mold or introduction of any kind of dirt had to be carefully avoided, in order to prevent the acquisition of pattern perturbations not originated by particles, that could confuse the training algorithm.

In addition, to be sure that the classifier could not classify the patterns on the base of the mixtures properties, the preparation of particles mixtures belonging to different classes had to be carefully performed in a reproducible way.

To learn how to take control over these critical effects and aspects, many different tests and practical investigations had to be carried out. To this end, the use of optical microscopes was essential to visualize the mixture and particles. Here we list the main improvements made concerning the particle mixtures:

- Mold formation and accumulation of dirt in the mixtures was prevented by dissolving a small dose of water purification tablets (from the Oasis brand) and by limiting the reuse of mixtures to a maximum of two weeks.

- Particles with larger difference in diameter were used (13.5µm and 17.3µm). Given that the nominal standard deviation of the diameters is around 3%, this helped to visually distinguish the particle types with the microscope. Moreover, this simplified the classification task, which was still far from being properly learned by the classifier.

- A microfluidic channel with larger cross section (100µm $\times$ 100µm) was employed. This was necessary to avoid too frequent clogging of the fluidic circuit. A drawback of this choice was that the particles could undergo larger displacements in the transverse directions w.r.t. the flow, increasing the difficulty of the classification task.

- After testing mixtures with various amounts of particles and surfactant, an optimal combination of the components' amount was determined: 30mL of deionized water, 18.5µL of original particle mixture (with 5% solid content) and 18.5µL of surfactant.

- The surfactant solubility was improved by warming up the water in which it was dissolved.

4.2.2 Noise, background subtraction and background pattern detection

One of the main issues arising from the analysis of the results obtained using the initial setup, was that the SNR of the recorded particle signal was too low. Further investigations indicated that the recorded patterns after background subtraction showed evident distortions and fluctuations during the measurement. Typically, these had a characteristic time of several frames and were ascribed to small changes in the laser beam propagation due to vibrations and to fluctuations of the laser intensity. This type of noise was responsible for a poor SNR in background-subtracted patterns and was sometimes so intense that its peaks were mistaken for particles by the employed background detection algorithm.

In order to diminish these unwanted effects, a more suitable background subtraction method was developed. In particular, instead of subtracting from each image the first recorded image, we subtracted an image obtained by averaging the previous 20 patterns and we calculated the absolute value of the pixel differences. This allowed to remove the influence of not too fast fluctuations in the acquired patterns. Examples of acquired patterns are shown in Fig. 4.6. It should be specified that, since the considered classification algorithms include the normalization of the values at each pixel (feature scaling), only relative values w.r.t. other pixels are relevant.

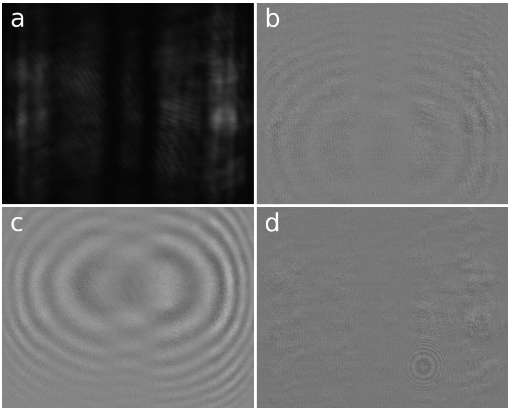


Figure 4.6: Examples of acquired patterns using the setup configuration without SLM (Fig. 4.5) and employing an improved background subtraction method based on the moving average of acquired images. *a*: Background pattern before background subtraction. *b, c*: Particle samples after background subtraction. It can be noticed that the particle pattern can have a different intensity and center position in the acquired images, depending on the particle location in the channel at the moment of the acquisition. *d*: Background sample after background subtraction. The small concentric pattern at the bottom right of the image is probably due to a dust particle passing through the laser beam. The rest of the visible pattern is ascribed to the noise generated by vibrations of the optical components.

Moreover, also the background detection algorithm had to be improved, so that we could be sure that only patterns perturbed by the presence of a particle would be considered as particle samples. The search for a suitable algorithm was not trivial and required significant effort. We tested different implementations based on the following detection methods:

- Threshold applied to moving average of auto-correlation values (see Eq. 2.1) calculated on different image regions.

- Threshold applied to *Fast Fourier Transform* (FFT) values calculated on different image regions.

- Machine learning classification using combinations of features extracted through sum of pixels along columns or rows, auto-correlation function, FFT, and *linear discriminant analysis* (LDA).

The machine learning approach was not successful due to the difficulty in manually labelling enough background and particles samples for the training algorithm. Eventually, the moving threshold method using FFT calculated on specific image regions was chosen at this stage, because it provided a satisfying accuracy.

Since the final goal of this experiment was to develop a computationally cheap particle classification process, it would not have been convenient to employ such a relatively complicated background detection algorithm as part of the actual classification inference. Indeed, at this stage background detection was meant not to be part of the inference process, but just to label background images for the classifier training, which would consider backgrounds as a separate class. However, as we will see in Section 4.4, in the final version of the setup the SNR was considerably increased, bubble generation was avoided and the particle mixtures were not reused, preventing the introduction of dirt in the fluidic circuit. These improvements allowed to employ a much simpler and more effective background detection algorithm (see Subsection 4.4.1), which was then incorporated in the inference process, leaving the machine learning classifier with a simpler two-class problem.

4.2.3 Detection and solution of the bubble generation problem

In order to investigate the presence of unwanted sources of pattern perturbation other than the PMMA particles, measurements using a reference solution without particles were performed. This showed that patterns similar to the ones projected by the PMMA microparticles were abundant in the recorded images. To check whether dirt in the solution were the cause, a powerful optical microscope was employed (images similar to Fig. 4.3 were obtained). However, the employed solution appeared relatively clean.

After a closer examination of the measurement process, it was observed that bubbles were being generated at the interface between the input tube and the microfluidic channel. Indeed, the recorded patterns were projected by rapidly flowing bubbles which produced similar but stronger perturbations compared to the PMMA particles (see Fig. 4.7, to be compared with Fig. 4.6 *b*, *c*). The stronger intensity of the bubble patterns is ascribed to the much larger refractive index contrast given by air w.r.t. PMMA.

Afterwards, we found that this issue could be avoided by pushing the mixture in the fluidic circuit with the syringe, instead of pulling it. An intuitive explana-

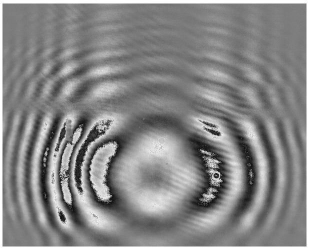


Figure 4.7: Examples of acquired pattern projected by a rapidly flowing bubble, after background subtraction. The pattern perturbation is similar but stronger w.r.t. the one caused by PMMA particles (compare with Fig. 4.6 *b*, *c*).

tion is that this prevents air from being sucked in at the connection of the tube with the channel. Nevertheless, this solution implied that the particle mixture had to be initially contained in the syringe, where the magnetic stirrer could not be used. Therefore, while the syringe was actuated by the syringe pump, in less than a minute the particles sunk to the bottom and thus they stopped entering the fluidic circuit. To overcome this problem, the syringe pump was not used anymore, and the syringe was manually actuated and kept in an almost vertical position. This allowed to inject the sinking particles for a longer time and to easily shake the mixture in the syringe. Fortunately, the obtained flow rate was steady enough for this specific application. Indeed, the short exposure time of the camera allowed to acquire patterns that were not heavily affected by blurring, limiting the degree by which the particle flowing velocity affected the measurements.

4.2.4 Intertwined class measurements to study and prevent measurement bias

In order to obtain training and test samples that would not cause measurement bias, chronologically *intertwined class measurements* were performed. That is, a measurement session using particles of class A was performed, then a measurement session using B particles, then a session with A particles, then with B particles, and so on. In fact, this is an attempt to break the correlation between slow drifts of the measurement conditions (that cause corresponding changes of noise in the acquired patterns) and the class labels. Intuitively, employing this measurement approach should prevent the training algorithm from easily learning the classification task from the noise in the images, since there is no more a class that is measured before another. Indeed, the training algorithm is forced to focus on the actual particle signal and to learn the classification by generalizing over the different measurement conditions that characterize the training samples. Moreover, testing the trained classifier on test samples measured under different conditions, produces a more general evaluation of the classification performance. A demonstration of the benefits of performing intertwined class measurements is provided in the next chapter.

The method was applied by performing 9 measurement sessions per particle class, each providing around 10,000 images, from which around 500 particle patterns and around 500 background patterns were selected. Obviously, many more background patterns were available from each session, but we opted to employ a similar number of samples for each case. The 18 measurement sessions were spread over a period of 2 weeks.

4.2.5 Investigating the source of measurement bias

Another advantage of intertwined class measurements is that the source and some properties of measurement bias can be investigated by employing only background patterns to train and test the classifier. Let us see here how this was done in this specific case. For the sake of simplicity, let us call *background A samples* and *background B samples* the samples labelled as background originated from a measurement session where particles of respectively class A and class B were used. And let us call *background classification* the task of distinguishing background A samples from background B samples, i.e. considering corresponding background samples instead of particle samples.

In order to keep both overfitting and the computational cost of training under control, the recorded images were not directly employed as samples. Instead, feature extraction methods similar to the one applied to the cell hologram classification described in Chapter 2 were considered. I.e., *1D samples* were obtained by summing all the pixel values along the rows of the 2D images, while *FFT samples* were obtained by applying the fast Fourier transform (FFT) to the 1D samples. This allowed to represent the particle information in a space with much lower dimensionality (i.e. the samples have much less features) w.r.t. full-resolution 2D images. In particular, starting from around one million pixels, around one thousand features were extracted. It should be stressed that feature extraction was employed at this stage with the purpose of testing the classification feasibility by employing a familiar and more intuitive machine learning approach. Afterwards, feature extraction was not considered anymore, in order to keep the computational cost of the classification low.

Let us now describe some background classifications (i.e. where the samples do not contain particle information) performed to investigate measurement bias in the intertwined class measurements. At first, background classification was evaluated employing the background samples from 8 measurement sessions per class for training, and the background samples from 1 measurement session per class for testing. This was repeated for different permutations of the measurement sessions so that a 9-fold cross-validation was performed. Therefore, in each classification, the test samples did not share the measurement session with any training samples. This means that the only connection (that could be exploited by the training algorithm to learn the classification of test samples) between the training and the test samples was the fluid part of the particle mixtures. Simply put, this classification process could provide high test accuracy if and only if the classifier were able to exploit differences between mixtures (other than the particles size) to learn the classification task. If this were the case, a higher accuracy in the mixture preparation would have been necessary, possibly undermining at the root the feasibility of the whole experiment.

Fortunately, the obtained test accuracies were $(55 \pm 10)\%$ and $(55 \pm 23)\%$, respectively for the use of 1D samples and of FFT samples, where the error is given by $\pm$ one standard deviation. These accuracy estimations are comparable with the ones obtained through random choice, meaning that the classifier could not learn the classification task at all. It should be stressed that this test was necessary to trust any conclusion derived from any of the classification evaluations presented in this chapter. Indeed, if this test were not performed, it would be impossible to know whether the success of subsequent classification processes had to be ascribed to differences in the fluid content of the mixtures.

To test the presence of measurement bias due to drift of measurement conditions, the classifier was trained using 9 out of 10 background samples from every measurement session and it was tested on the remaining background samples. That is, the training and the test samples sets both contained samples originated from all 18 sessions. This means that, in order for the classifier to successfully learn the classification, it had to group the samples coming from the 18 sessions in two class groups, on the base of changes in measurement conditions. Since the classifier is linear, this would require the existence of a hyperplane in the feature space that could separate these 18 clusters of points according to the two classes. However, the continuity of the measurement conditions drifts w.r.t. the class labels was broken by the intertwined class measurements, and therefore this background classification task might seem quite complicated for a simple linear classifier. Nevertheless, given the relatively high number of features (around one thousand), a successful classification would be possible if the representation of the measurement conditions changes had a high enough dimensionality in the feature space. In other words, the requirement is that the drift in measurement conditions influenced several features so that they were scarcely correlated. Or, even more simply put, this influence was rich and diverse enough. For example, a simple drift in overall image intensity (provided that the extracted features were intensity dependent), in this case could not be exploited to learn the classification. Instead, for a successful classification, at least every measurement session should be characterized by a unique combination of pixel values, and these combinations should be represented by the extracted features so that the two classes are linearly separable (see Chapter 1).

Even though these requirements might seem unlikely to be met in this experiment, surprisingly the background classification was relatively successful, with errors of $(12 \pm 1)\%$ and $(10 \pm 1)\%$ respectively for 1D samples and FFT samples. This shows that this kind of flow cytometry implementation is extremely prone to be affected by measurement bias. Still, our intuition was that this elusive type of overfitting could be significantly reduced by reducing the noise in the recorded images. In particular, when particle samples are considered instead of backgrounds, the SNR is a key property. Indeed, if the particle signal is strong enough in the patterns, it would be more likely that the classifier finds it easier to learn the task on the basis of the actual particle characteristics rather than focusing on the measurement conditions. In the light of these arguments, the improvement of the SNR was one of the main goals of the next steps in setup development, which are described in Section 4.3.

4.2.6 Attempts to bias-free particle classification

At this stage in setup development, several particle classification attempts were tried without the scattering operation of the SLM. In particular, 8 different initial sets of features were extracted in different ways from the available patterns. In short, the 8 feature extraction methods consisted of generating 1D samples and FFT samples, starting from the full original pattern or from specific regions of the images, with and without background subtraction. The linear classifier was trained and tested on particle classifications and corresponding background classifications employing the 8 extracted feature sets one by one. Moreover, additional classifications were attempted by further applying feature selection or extraction (using LDA) on top of the 8 feature sets in order to lower the number of considered features. As previously done, a 9-fold cross-validation was applied, where the folds contained samples from separated and intertwined measurement sessions.

We do not go further into the details of these attempts, since they all provided similar results. In summary, relatively high training accuracies were obtained in particle classification, but the test accuracies of the corresponding background classifications reached very similar values. This suggested that the classifier was mostly learning the classification on the basis of the background information. Such a conclusion was confirmed by the fact that test accuracies of around 50% were obtained in the attempted particle classifications, which at least demonstrated that the intertwined class approach allowed to detect the poor generalization learnt because of background bias. From these results we concluded that measurement bias needed to be significantly reduced, and that the SNR in the produced patterns had to be substantially improved, before we could demonstrate any reliable particle classification.

The study described in this section, where a test setup without SLM was considered, provided us with a better knowledge of the measurement process and related issues, and allowed us to significantly improve the experiment. However, at this point we preferred to resume the employment of the SLM, so that further required improvements could be made while developing the full experimental configuration where scatterer layers can be added to the optical path.

4.3 Improved experiment with SLM

In this section we describe how the particle classification using an SLM was attempted, exploiting the information and the improvements obtained through the work presented in the previous sections.

4.3.1 Two patterns on one camera

In Section 4.1 we have seen that the beam directly projected by the microfluidic channel and the one that was additionally reflected by the SLM were measured by two different cameras (see Fig. 4.1). We call the corresponding acquired patterns *direct pattern* and *SLM pattern* respectively. This configuration required that the two cameras were programmed to work in a synchronized way, limiting the achievable frame rate. However, these limitations implied that longer measurements had to be performed in order to obtain enough samples. Because of the measurements improvements introduced afterwards (presented in Section 4.2), it became less practical to perform such long measurements. In particular, the magnetic stirrer could not be used anymore to prevent the particles from sinking before being introduced in the fluidic circuit, limiting the duration of a measurement session. Moreover, the adoption of the intertwined class measurement method greatly increased the amount of time and effort to perform a full set of measurement sessions.

In order to speed up the acquisition of enough particle patterns, the setup was modified so that a single camera (the faster Ximea CMOS sensor) acquired both direct patterns and SLM patterns at the same time (see Fig. 4.8). To ensure that the two patterns would not overlap, a small black panel was attached to the camera so that it divided the screen in two regions. Moreover, since SLM patterns had significantly lower intensity, the intensity of direct patterns was lowered using an optical attenuator in order to roughly match the intensities of the two acquired patterns. Because of the relatively high resolution of the image sensor (1280 x 1024 pixels), both patterns could be measured in more than sufficient detail.

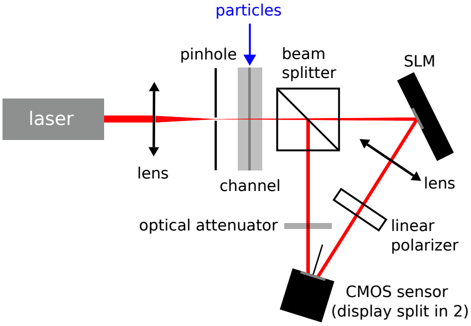


Figure 4.8: Both direct patterns and the SLM patterns are acquired by a single camera, whose screen was separated by a small panel. An optical attenuator was introduced to ensure that the two acquired patterns had similar intensities.

4.3.2 Noise generated by the SLM and optimization of the scatterer configuration

In order to improve the SNR in the measured patterns, the measurements employing this setup version were performed in a dark environment, with lights off. Moreover, it was discovered that the SLM had to be configured in a different way w.r.t. how it was reported in the manual (the SLM was meant to work with infrared light), in order to maximize the contrast of the projected pattern when using red light. In particular, the value range of an SLM pixel that spans from 0 to 2π phase shift had to be adjusted from $[0, 254]$ to $[178, 254]$. Indeed, as one can expect, the phase shift applied to visible light by different liquid crystal orientations is higher than the one applied to infrared light. An example of the employed SLM phase shift pattern and the corresponding camera image containing the SLM pattern and the direct pattern are shown in Fig. 4.9 *a, b*.

After background subtraction, the particle signal was clearly visible in direct patterns (see Fig. 4.9 *d*, *f*) and it was also easily detected by the background detection algorithm. However, it was much more difficult to detect the particle presence by naked eye from the SLM patterns because of the noise generated by some fast changes in the projected pattern, which could not be removed well by background subtraction (see Fig. 4.9 *c*, *e*). We ascribed this noise mainly to phase flicker in the SLM. By randomly selecting a single pixel from the noisy regions in the recorded SLM patterns and by visualizing its intensity over time, we could see

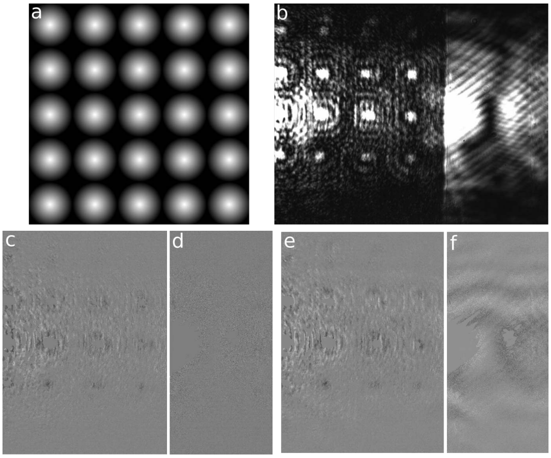


Figure 4.9: *a, b*: Examples respectively of a SLM phase shift pattern and the corresponding acquired image, containing both the SLM pattern (left) and the direct pattern (right). The overlapping of the two patterns was avoided by dividing the screen in two regions with an attached panel. *c, d*: Background patterns from the same image, after background subtraction. *e, f*: Particle patterns from the same image, after background subtraction. It can be noticed that while the presence of the particle is clearly visible in the direct patterns, in the SLM patterns it is partially hidden by noise mainly ascribed to phase flicker in the SLM.

that the noise appears as an overlap of different dynamics: a slow drift, a step-like trend, periodic fluctuations and apparently chaotic fluctuations, each at different time scales (see Fig. 4.10). Moreover, these dynamics seemed to quantitatively differ from pixel to pixel. Compared to simple white noise, this type of noise is particularly difficult to treat and could easily generate measurement bias. The fact that it showed an intensity comparable to the particle signal in background-subtracted images did not seem promising at all. In spite of this, given the central role of the SLM in this experiment, we decided to attempt particle classification employing this setup configuration.

Inspired by the previous proof-of-principle of cell classification based on FDTD simulation (see Chapter 3), we designed 12 different phase shift patterns to be actuated by the SLM (see Fig. 4.11). The goal was to estimate which of these SLM configurations would provide the strongest classification improvement. However, performing proper intertwined class measurements recording thousands of particle patterns for each SLM configuration would have required

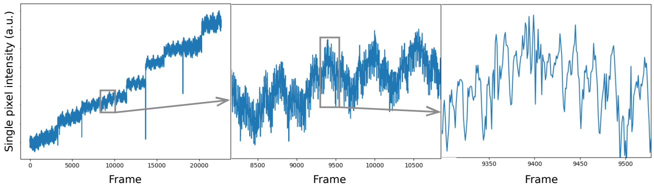


Figure 4.10: Time evolution of a single pixel randomly selected from the noisy regions in the recorded SLM patterns. The second and the third plots represent a zoomed version of the plot at their left. It can be noticed that the noise appears as an overlap of a slow drift, a step-like trend, periodic fluctuations and apparently chaotic fluctuations, each at different time scales

an excessive amount of time and effort. Therefore, we limited ourselves to quick measurements of around 100 particle samples per class and per configuration, that were not sufficient for classification. Instead, we aimed to roughly evaluate the goodness of each configuration by considering the average overall image contrast caused by particles w.r.t. the background. Unfortunately, because of the noise introduced by the SLM and the relatively low number of particle samples, it was not possible to also obtain a reliable estimation of the contrast between classes. Among the explored configuration, the 4^{th} was the one that provided the highest average overall contrast between particle and background samples, and it was therefore selected for a full intertwined class measurement.

4.3.3 Attempt to particle classification using SLM patterns

Employing the selected SLM configuration, an intertwined class measurement of 8 sessions per class was performed. Each session provided around 12,000 samples and a total of around 5,000 particle patterns were collected. It should be stressed that the goal of the experiment was to demonstrate that SLM patterns were easier to classify than direct patterns, when a linear classifier was directly applied to the pixel values. Therefore, in order to keep the computational cost low, similarly as in Chapter 3, no feature extraction method which generated new features was considered. However, given the very large number of available features (more than 1 million pixels) and the fact that most of these features were expected not to convey information about the particle class but just noise, it was necessary not to employ them all but to select a very small number. In particular, considering the number of available particle samples, we aimed to eventually select around 100 or less features for the linear classifier, in order to avoid overfitting. Moreover, it should be noticed that employing only few selected pixels would imply a smaller computational cost of the classification inference, and it

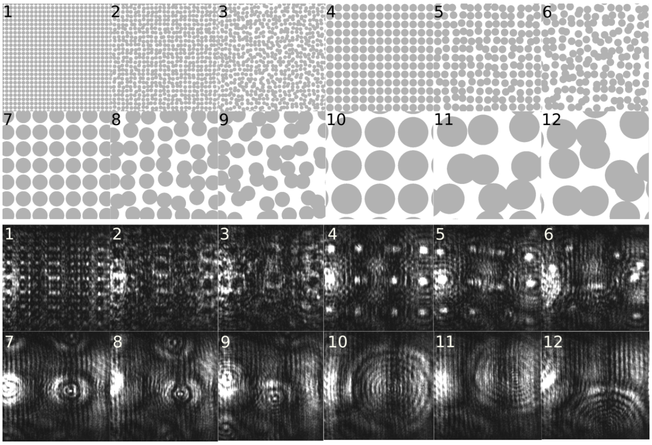


Figure 4.11: *Upper figures:* The 12 explored phase shift patterns actuated by the SLM. *Lower figures:* Corresponding projected patterns (background samples).

could possibly provide a further advantage because of an increase in the frame rate of the camera.

Designing a suitable feature selection algorithm for this application was not trivial. In doing this, we had multiple goals in mind. First of all, we aimed to discard the noisy or irrelevant pixels, which were expected to be the vast majority, in order to select the features with high enough *relevance* for the targeted task. However, two types of noise were to be distinguished and addressed separately:

- The most common type of noise (here referred to as *neutral noise*), which is not correlated with the class labels. In machine learning this is usually treated with validation procedures (such as k-fold cross-validation) in order to control overfitting.

- A more elusive type of noise (here referred to as *biasing noise*), which is correlated with class labels and may lead to measurement bias. This has to be addressed by exploiting the advantages of having performed intertwined class measurements.

Moreover, noise is not the only issue to be aware of. Indeed, because of the high resolution of the acquired images w.r.t. the patterns details, it is likely that groups of adjacent pixels convey highly correlated information regarding the particle class. In other words, they provide redundant information. Avoiding the selection of many redundant features is important to reduce overfitting and the complexity of the classification inference. But most of all, it is critical in order to avoid that non-redundant pixels that convey important alternative information about particles are left out from the selection, because a lot of relevant but redundant pixels were selected first. This is a well known problem which can be addressed with the popular *minimum–redundancy maximum–relevance* heuristic approach [3]. A trade off between redundancy and relevance can be optimally and intrinsically achieved in powerful feature selection methods based on direct evaluation of the classifier performance on different feature subsets (*wrappers*, see Chapter 1). However, these methods require exceedingly high computational resources when many features are to be evaluated in the selection. Therefore, in our case, we could apply a wrapper only after most of the available pixels were already discarded by computationally cheaper feature selection steps.

Filter-based feature selection refers to heuristic scoring methods that usually rely on the calculation of computationally cheap statistics in order to select a feature subset with certain characteristics. We chose this approach for the first feature selection steps, which had to deal with a large number of features (in Fig. 4.12 we show examples of the pixels selected by three of such steps). Only in the last step (to pass from 1000 pixels to around 100 selected pixels) we employed a wrapper based on the cross-validated accuracy obtained by the linear classifier. Several attempts were made to discard pixels affected by neutral and biasing noise, and to select non-redundant features. However, the biasing noise in the SLM patterns was still too strong and we could not avoid measurement bias. Therefore, we could not achieve good accuracies when the classifier was tested on samples from a measurement session that was not considered for training. Since these attempts were not successful and we are not yet describing the final setup and measurements, here we do not go into more detail. Finally, we reached the conclusion that the noise introduced by the SLM flicker was unfortunately too strong and we started looking for an alternative way to implement suitable optical scatterers.

4.4 Experiment with physical scattering media

In this Section we describe tests and improvements regarding the setup and the measurements, which resulted in the final realization of the experiment described in Chapter 5. In particular, among other changes, we substituted the SLM with physical scattering layers (such as diffraction gratings) and we simplified the setup so that it generated only one pattern at a time through a single optical axis.

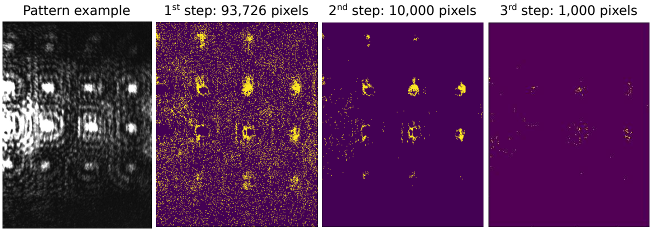


Figure 4.12: *Left:* Example of recorded SLM pattern to which the feature selection was applied. *Right:* Corresponding selected pixels respectively at 3 consecutive steps (using filter-based methods) of feature selection. The selected pixels are mainly gathered around the luminous spots in the pattern, projected by the SLM. Unfortunately, these are also the regions that are most affected by the noise due to SLM flicker.

4.4.1 Setup changes and improvement of SNR

In the previous Section we have seen that the SLM introduces too much noise in the projected intensity pattern. Thus, we decided to substitute it with simpler and less noisy transmissive scattering layers such as holographic diffraction gratings and transparent microparticles (further details will be given later on). These layers introduced a additional advantages w.r.t. the SLM: they did not require light polarization and they generally provided lower optical losses. Since a more intense laser beam could reach the camera we did not need to perform measurements in a dark room anymore. More importantly, because of the lower optical losses we could afford to use a smaller pinhole (with diameter of $25\mu\text{m}$) that was clamped to the slide containing the microfluidic channel, so that it could be as close as possible to the flowing particles. Such an improvement has been critical in reaching a sufficiently high SNR in particle patterns, thanks to the fact that a smaller region of the slide containing the channel was illuminated. Indeed, this reduced the acquired background pattern intensity (and the noise conveyed by it) w.r.t. pattern perturbations projected by particles.

However, a significant amount of noise was still originated from vibrations and movements of the optical components determining the optical path of the laser beam. In order to reduce this noise as much as possible, we moved the setup on a suspended table and we fixed all the physical connections that could convey movements or vibrations to the setup, such as the camera cable or the silicone tube that connected the syringe with the microfluidic channel. For the same reasons, it proved to be beneficial to clamp as many components as possible together, minimizing the use of mounts fixed to the suspended table. To do so, we asked our laboratory technician Peter Guns to create (with a 3D printer) a customized mount that would fix the employed scattering layers in front of the camera, at a distance of choice. Also the pinhole was directly clamped to the microfluidic channel slide and did not require a dedicated mount anymore.

The achieved improvement in SNR allowed to record a higher number of particle patterns for a given measurement duration. The reason is, as we will discuss in detail in the next chapter, that the strength of pattern perturbations due to particles intuitively depends on whether a particle was well centered or not w.r.t. the laser beam at the time of image acquisition. A higher SNR implies that particles that were not well centered, and whose pattern perturbation would have been covered by the noise in the previous setup configuration, could now be detected by the background detection algorithm. Therefore, shorter measurement sessions could provide enough particle samples, which in turn lowered the probability of clogging the channel and generally reduced the time and effort required by measurements. Additionally, a significantly smaller quantity of particles per measurement session was used, which allowed to avoid reusing the particle mixtures. Because of this we could almost completely prevent the presence of accumulated dirt in the flowing fluid, removing another relevant source of noise affecting the acquired patterns.

Thanks to these enhancements, we did not need to acquire two different patterns (i.e. direct patterns and the ones projected by the scattering layers) at the same time. Indeed, the main reason why we needed to record also direct patterns, was that after background subtraction the particle signal could be more easily distinguished from the noise by considering the shape of pattern perturbations. Therefore, the direct patterns were used to distinguish a background pattern from a particle pattern through the background detection algorithm. In this case, however, the increase in SNR guaranteed that large number of pattern perturbations due to particles had a much higher overall intensity than the perturbations due to noise. Thus, we could employ a much simpler and computationally cheap background detection algorithm based on the pattern intensity after background subtraction (that will be described in the next chapter), which could be applied also on the patterns projected by the scattering layers. Because we did not need the acquisition of direct patterns anymore, we could employ a much simpler setup with a single optical axis (see Fig. 4.13 *a*). In this configuration we could finally acquire patterns where the particle signal was much stronger than the noise, after background subtraction (e.g. see Fig. 4.13 *c, d, e*).

It should be stressed that the saturation of some pixels in the central region of the acquired patterns is not considered an issue. Instead, it allows to exploit the dynamic range of the camera to measure the weak light impinging on the sensor regions far from the center. Then, by adjusting the distance of the camera from the microfluidic channel, the field of view can be expanded or shrinked, in order to center the most useful components of the measured scattered light w.r.t. the dynamic range.

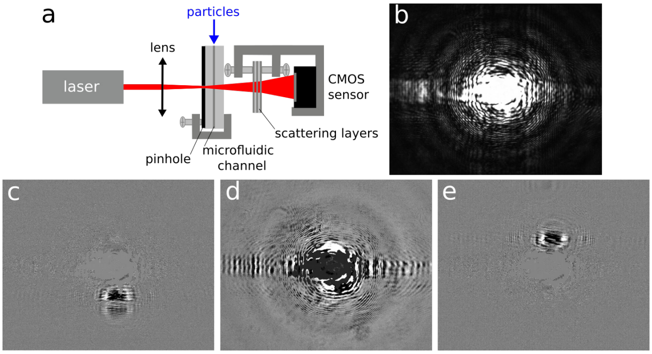


Figure 4.13: *a:* Drawing of the setup with a single optical axis. The pinhole is clamped to the slide containing the microfluidic channel, in order to reduce noise from vibrations and to increase the relative intensity of the particle signal w.r.t. the background illumination. The scattering layers are fixed to the camera using a mount that allows to adjust their number, rotation and distance to the camera. *b:* Example of acquired pattern without scatterers. *c, d, e:* Examples of patterns after background subtraction. As can be noticed, strength and shape of pattern perturbation due to particles strongly depend on the position of a particle w.r.t. the illumination center at the time of image acquisition.

4.4.2 Testing different types of scattering layers

Even though we had to renounce the high programmability of the scatterer configuration provided by the SLM, this was partially compensated by the possibility of changing the number, type, rotation and distance (to the camera or to the channel) of the employed physical scatterer layers. In addition, also the effects depending on the exposure time of the camera were explored. Indeed, longer exposures enhanced the least luminous details of the acquired patterns, while the most intense could be excluded through pixel saturation. Several combinations of these parameters were tested for each of the different scatterer layers that we describe in this Section (some examples of obtained interference patterns are shown in Figs. 4.14, 4.15 and 4.16).

As an example, at first we employed the same PMMA microspheres that we aimed to classify as dielectric scatterers. To do so, the transparent microparticles were sandwiched between two transparent microscopic slides made of glass. The slides were then fixed together by applying superglue along the sides. The obtained scattering layer projected a random and granulated distortion of the direct pattern projected by the channel and particles, so that this could still be recognizable (see Fig. 4.15). By increasing the distance between the chan-

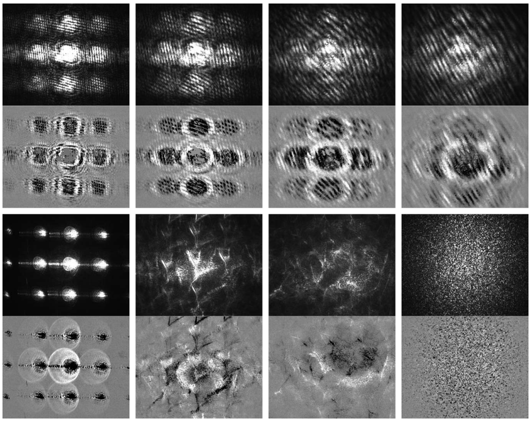


Figure 4.14: Examples of acquired particle patterns before (with black background) and after (with grey background) background subtraction. Double axis diffractive gratings, a lens and diffusive media were employed in several combinations and configurations to generate different types of interference patterns.

nel and the scattering layer, the pattern distortion due to the scatterers became more and more finely granulated. As opposed to this kind of scattering layer which distorts the background and particle patterns only partially, a diffusive medium completely disintegrates the original pattern structure (see Fig. 4.14 bottom right). The employed scattering layer that most resembles the dielectric structures considered in the numerical proof-of-concept presented in Chapter

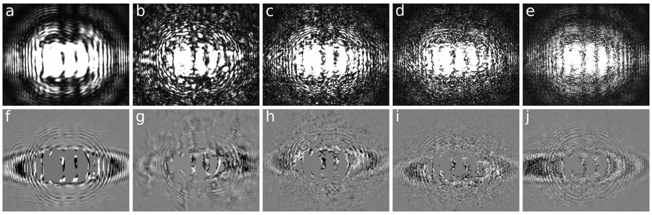


Figure 4.15: Patterns acquired using different scatterer configurations before (*top*) and after (*bottom*) background subtraction. From left to right: without scattering layer (*a, f*) and using PMMA microparticles sandwiched between two glass slides, at a distance from the microfluidic channel of 0.2 cm (*b, g*), 1.1 cm (*c,h*), 2.1 cm (*d, i*) and 3.7 cm (e, j).

3 is a transmissive diffractive grating. In particular, we employed double-axis holographic diffraction gratings (with a line period of around 1.88 µm), which projects several similar patterns (corresponding to the diffraction orders) that are approximate copies of the original pattern (see Fig. 4.16, first and second rows). The spacing of the acquired pattern copies depends on the distance between the diffraction grating and the camera. Therefore, by tuning such a parameter, the pattern copies can be more or less overlapped, enabling the interaction of the original pattern with different copies of itself in different regions.

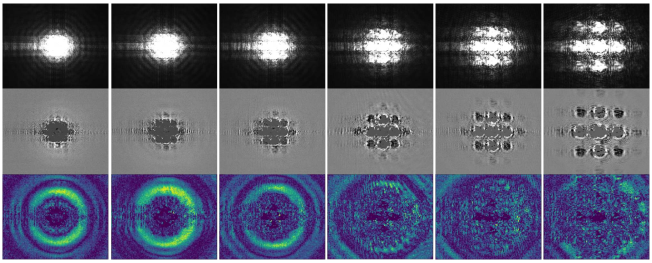


Figure 4.16: Particle patterns obtained employing a double-axis transmissive diffraction grating, fixed at different distances from the camera (from left to right, respectively 7, 7.5, 8, 9, 10 and 12 mm). The patterns before and after background subtraction are shown in the first and second row. In the last row are the corresponding colormaps of the class separation measure for each pixel in the images. The pixels with lighter colour show higher class separation, visually indicating the pattern regions that are most relevant for the considered classification task.

4.4.3 Attempt to predict classification outcomes from few samples

Because of the substantial time and effort required to generate the samples for the full training and testing of the classifier, we tried to find an alternative way to evaluate how beneficial a certain scatterer configuration could be for the classification. In particular, in order to evaluate tens of different possibilities, we aimed to develop a measure for the performance of the optical configuration that required few samples (hundreds instead of thousands) and few sessions of the intertwined class measurement (only two per class). As we discussed in Chapter 2, measuring the separation of the feature value distributions between different classes provides an estimation of the relevance of that feature (e.g. a pixel) for the considered classification task (named *class separation*). On the other hand, if the same evaluation is performed considering particle samples from the same class but from different measurement sessions, we could obtain a measure of how much the given feature might contribute to measurement bias (named *session separation*). Therefore, the scatterer configurations that overall provided pixel values with high class separation and low session separation could be considered as good ones. However, the previously employed *similarity measure* (Eq. 2.4) is based on the assumption that the feature values approximately follow a normal distribution. In our case, it would be sufficient that these distributions were central, i.e. symmetric w.r.t. to a center. Still, the measured intensity at a pixel is proportional to the absolute square of the impinging complex optical signal, and thus it is likely that the feature distributions are skewed when close enough to the zero (e.g. see first plot in Fig. 4.17). Because of this, we needed to consider a relevance measure for each pixel that could be suitable for skewed distributions.

For these reasons, we considered a nonparametric approach, i.e. based on statistics that do not assume that the studied values follow a specific distribution. In particular, we employed a statistic (called here *class separation measure*), that is robust against outliers and non-normality, based on the *Mann–Whitney U* statistic [4]. Given a pixel, the class separation measure tells how stochastically larger or smaller are the values corresponding to one class w.r.t. to the ones belonging to other classes (e.g. see Fig. 4.17). In particular, the U statistic is calculated as follows: given an observation from one set (i.e. a feature value from one class), count how many times this observation is larger than the observations in the other set (i.e. the other class). The following normalized (from 0 to 1) expression was considered:

$$\frac{|U - (n_A n_B + 1)/2|}{(n_A n_B + 1)/2} \tag{4.1}$$

Where U is the aforementioned statistic, calculated through the Python function *scipy.stats.mannwhitneyu* (with the “alternative” parameter set to “two-sided”); n_A and n_B are the number of samples belonging to class A and B respectively. To measure the relevance of the patterns obtained using a certain scatterer configuration, we considered the mean square of the class separation calculated on the 20% of the pixels with highest class separation. Note that we chose to measure the overall relevance of a scatterer configuration only using the most relevant pixels, in order to exclude the influence of noisy pixels and to advantage the configurations that allowed to use a smaller number of features for classification.

As we previously discussed, relevance is not the only important parameter to heuristically predict the usefulness of a set of features, or, in this specific case, of a given scatterer configuration. The redundancy of the information conveyed by the pixels had to be taken into consideration as well. As a redundancy measure

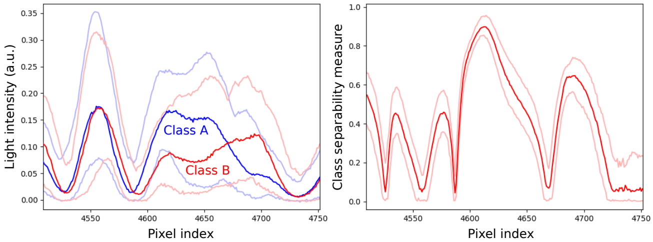


Figure 4.17: Example of class separation measure (*right*) calculated from pixel intensity distributions corresponding to two different classes (*left*). The central darker lines correspond to the median of the value distributions represented by the plots, while the faint lines show the corresponding 95% percentile range. For each pixel, the class separation measure tells how stochastically larger or smaller are the values corresponding to one class w.r.t. to the ones belonging to other classes. Therefore it can be used as a measure of the relevance of a certain pixel w.r.t. the classification task.

we considered the mean square of the nonparametric Kendall correlation coefficients between the pixels with the highest relevance (the 20%, 10%, 5% and 1% of the most relevant pixels were considered). Kendall's τ coefficient is a correlation estimation in a way similar to the Mann-Whitney U. It is the normalized difference between concordant and discordant pairs of observations. For example, considering the pair of feature values from the samples i and j, the corresponding pairs are concordant if in both classes we have: feature$_i$ $>$ feature$_j$ or feature$_i$ $<$ feature$_j$.

In order to test a maximum relevance-minimum redundancy approach, we measured the relevance and redundancy of the 1D patterns obtained for different scatterer configurations in the proof-of-concept based on FDTD simulation presented in Chapter 3. From the combination of the two measures applied on 100 samples per class only, we could indeed qualitatively predict the difference in classification error between the cases with and without scatterers (see Fig. 4.18). Indeed, we can notice that the higher classification error obtained without scatterers (configuration A in the plots) is explained by a relatively high redundancy in the far field case, and by a relatively low relevance in the near field case. However, we could not accurately predict the slight differences between the results corresponding to different scatterer configurations, even by optimizing the prediction by building a linear combination of the two measures through linear regression.

A similar approach was also tested on different classification results obtained through intertwined class measurements with different scatterer configurations. The classifier was directly applied to the pixels of background-subtracted pat-

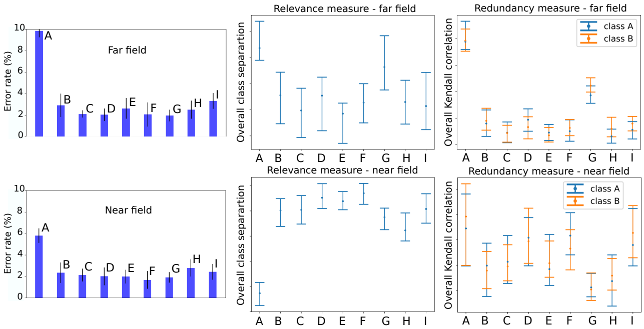


Figure 4.18: Overall relevance (*middle*) and redundancy (*right*) of the 1D patterns obtained for different scatterer configurations (labeled by capital letters) in the proof-of-concept based on FDTD simulation presented in Chapter 3, compared to the corresponding classification error (*left*). The first and second row of plots correspond respectively to the cases where far field and near field interference patterns were simulated. The relevance and redundancy measures appear to roughly predict the classification error for the different scatterer configurations, as according to the maximum-relevance minimum-redundancy paradigm.

terns and we optimized both L2 regularization and the pattern resolution (i.e. the number of pixels) through cross-validation (the classification pipeline is discussed in detail in the next chapter). However, in this case we had to take into account that the same biasing information causing measurement bias in the classifier training, could also bias in a similar way our heuristic prediction. Therefore, we considered both the *interclass* relevance and redundancy measures in relation to their *intersession* counterparts, where the classes were mixed. By intersession splitting of the samples, we mean that the samples are divided in two groups that do not share the measurement sessions, but contain both classes equally. This provides a baseline estimation representing the influence of measurement bias as disentangled from the influence of the class-specific information (which is averaged out in the gradient descent). Unfortunately, by employing the maximum relevance-minimum redundancy approach we could not obtain reliable enough predictions of the classification performance using only hundreds of samples. This difficulty was ascribed to the high dimensionality of the considered samples, and by the complex interplay between the overfitting caused by neutral noise and measurement bias caused by biasing noise. In fact, considering both the FDTD simulation results (Fig. 4.18) and the experimental data, we observed that just directly training a classifier on few samples, optimizing L2 regularization and image resolution, provided more reliable predictions than using our heuristic approach. This was indeed the method used to select the most promising scatterer configurations, so that they could be employed in longer measurements, which provided many more particle samples and comprised a higher number of class-intertwined measurement sessions.

Nevertheless, our class separation measure turned out to be a useful tool to visualize the pattern regions (or groups of pixels) that were most relevant for the classification task. This was done by plotting a colormap of the class separation of each pixel (e.g. see Fig. 4.16 bottom row). The obtained colormaps provided visual and intuitive hints on the relation between the scatterer configuration, the obtained particle pattern and the corresponding classification accuracy. For example, in Fig. 4.16 we can clearly see that when the diffraction orders projected by the transmissive grating were mostly overlapping at the center of the image, the relevant pixels were mostly arranged in rings outside the luminous center of the pattern. The same was true when no scatterers were used. However, as the diffraction orders gradually separated and spread over the whole image, these circular structures were disrupted and many smaller relevant regions appeared scattered all over the image. Interestingly, this did not systematically imply either a degradation or an improvement of the obtained classification performance.

4.4.4 Measurements and classification using 5 selected scatterer configurations.

After testing more than 40 different scatterer configurations through intertwined class measurements of various lengths, we selected the five most promising ones and performed longer measurements with those, in order to obtain reliable performance estimates. In particular, 12 intertwined class sessions per class were performed, each producing around 6500 images (a total of more than 150,000 images per scatterer configuration). These configurations, which we index with the following numbers for brevity, are:

1. No scattering layer, with a channel-camera distance of 7.5mm.
2. One diffraction grating, with channel-camera distance of 7.5mm and channel-grating distance of 6mm.
3. Two diffraction gratings, with same distances as in configuration 2.
4. One diffraction grating attached to the slide containing the channel, with channel-camera distance of 7mm.
5. As configuration 4, with channel-camera distance of 8mm.

In order to compare the estimated classification performance corresponding to the scatterer configurations, we chose to use box plots, which provide an informative yet simple representation of the distribution of the classification error estimates produced by the cross-validation cycles (a detailed description of the validation algorithm will be provided in the next chapter). Indeed, box plots allow us to take into consideration the average, the range, the skewness and the outliers of the represented error values for each configuration. In particular, boxes, whiskers, orange lines and green triangles represent respectively the interquartile range, the range, the median and the mean of the error values. The outliers (outer points distant more than 1.5× (interquartile range) from the interquartile range) are represented by circles.

The best performance was obtained by configuration 1, with a mean error of around 6% (see Fig. 4.19), which we considered reasonably low, indicating a successful classification. A slightly worse result was achieved by configuration 4. Therefore, the employed scatterer configurations did not show an advantage compared to the case without scatterers.

However, the experiment still had some flaws that prevented us from completely trusting this comparison. First, very different numbers of training samples were used for the different cases, respectively: 4124, 2091, 910, 3171 and 4401. This happened because of the employed background detection algorithm, which labels images as particle samples only if the overall perturbation intensity is higher than a threshold. Indeed, even if the measurements provided approximately the same number of images for each configuration, the number of available particle samples depended on the chosen threshold and on how the scatterer configuration affected the acquired particle perturbation intensity. The threshold was heuristically chosen by visually checking that the background detection algorithm did label as particle samples those image that were clearly affected by particle perturbation, rather than by noise. This means that the sample selection, and therefore the number and some of the properties of the samples obtained from a measurement, depended on the arbitrarity of human choice and on the SNR. Since the particle signal was significantly attenuated by the use of diffraction gratings, the SNR decreased with the number of employed grating layers, which explains the low number of samples obtained from the measurement with configuration 3. These issues, which projected uncertainty on the classification performance estimations and prevented reliable comparisons, were overcome in the final version of the experiment, described in the next chapter. The differences in number of samples are likely to have affected the results, especially noting that configurations 2 and 3 had also showed higher mean and variance in the error estimation. In machine learning, learning curves are usually plotted in order to investigate the relation between number of samples and classification performance. However, informative and comparable learning curves were difficult to create in this case, because of the complicated relation between the number of generated particle samples, signal intensity, scatterer configuration and number of intertwined class sessions. In the next chapter, we explain how

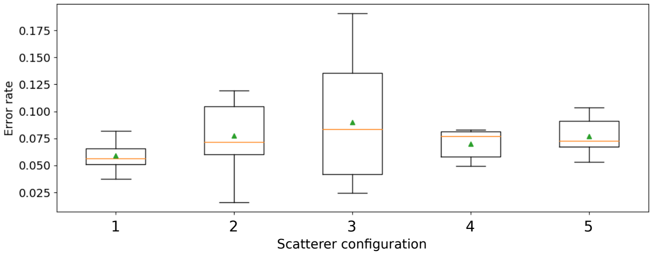


Figure 4.19: Box plot of the classification errors obtained employing the 5 selected scatterer configurations. The best performance (around 6% mean error) is given by configuration 1, which is without scatterers. These results were obtained with a camera exposure time relatively large considering the particle velocity, and therefore the particle samples were affected by motion blur.

we could disentangle these factors and obtain clear and comparable performance estimations.

In these measurements we employed a camera exposure time of 61µs. According to our calculations, this was bigger than the time a particle took to travel a distance equal to its diameter. This means that the acquired pattern perturbations created by the passage of particles was somewhat blurred due to particle motion, even if this was not immediately clear from visual inspection of particle patterns. This effect was likely to simplify the classification task from the readout classifier perspective, because the variability due to particle displacement along the channel was reduced. Indeed, we could not reach such low classification errors in subsequent measurements, when an effort to avoid motion blur was made. On the other hand, motion blur in the samples acquired in imaging flow cytometry is often an issue that limits the throughput [5]. In this context, it is interesting to note that the considered classification technique seems not to be negatively affected by this effect when microparticles are classified on the basis of the size. However, we still preferred to avoid motion blur in subsequent measurements, in order to limit the dependency of the particle signal on the flow velocity and also because we wanted to investigate whether scatterer use could provide an advantage in dealing with variability in particle displacement. Indeed, such an investigation would have been hindered if this variability were reduced or removed because of motion blur.

Thus, in the final measurements we considered a smaller exposure time of 29µs and we reduced the fluid velocity in the channel, to avoid motion blur. Moreover, in order to acquire more particle samples per measurement time, we increased the particle density in the mixture by around three times. For the final measurements (discussed in the next chapter) we considered the two best performing scatterer configurations 1 and 4.

4.5 Key practical aspects and possible setup improvements

We redeveloped the setup and the measurements described in this chapter from scratch, overcoming several problems and optimizing many different aspects. Although some of those aspects were mentioned before, here we list and summarize the most important practical details and requirements, which were often critical for the experimental realization.

- In our experience, shortcut learning (in the form of measurement bias) is very likely to happen, both due to the very high sensitivity of the type of measurements and due to the employed machine learning approach (in particular the application of a linear classifier on a large number of pixels). Intertwined class measurements with a sufficient number of sessions proved to be necessary. Moreover, it is important to maximize the SNR and to accurately prepare and treat the mixtures corresponding to different classes in the same way. A dedicated validation strategy has to be considered (e.g. see validation algorithm in next chapter).

- The pinhole should be as close as possible to the channel, and as small as possible, under the constraint that the laser beam that reaches the camera should be strong enough. This is important to acquire a particle signal that is intense enough w.r.t. the background signal.

- Noise due to vibrations should be minimized, as it can make the machine learning task much harder than it needs to be, and it greatly increases the risk of measurement bias. It is important to firmly clamp together as many of the components influencing the optical path as possible. A suspended table should be used and all the physical connections that might convey movement or vibrations to the system (such as the tube connected to the input of the microfluidic channel or the camera cable) should be firmly fixed to the table. A laptop, a magnetic stirrer or the arm or the pump that actuates the syringe should not be placed on the suspended table during measurements.

- It is important to try to measure under stable conditions, to reduce the risk of measurement bias. The laser might need to be switched on in advance to let it stabilize. The channel walls might slowly absorb water, which can change the refractive index of the PMMA and therefore the optical path.

Before starting to measure, we waited at least two hours after switching the laser on and after inserting water in the channel.

- We performed small reference measurements (using only water with no particles) at the beginning and at the end of each measurement. This was useful to check how the measurement condition changes had affected the acquired patterns, whether there was dirt in the water or in the fluidic circuit, and if bubbles were forming.

- It was necessary to avoid generation or insertion of bubbles inside the fluidic circuit. Pulling the fluid with the syringe, instead of pushing, favored bubble generation. Moreover, after the syringe actuation had started, it was important to wait the necessary time before starting the image acquisition, so that the air or the water initially present in the fluidic circuit was removed.

- We paid particular attention to avoid the contamination of measurement sessions corresponding to a certain particle class, with particles from other classes or with dirt. We used different syringes for different classes of particles and for the flushing liquid. We flushed the fluidic circuit before switching from one class to another. In particular, we flushed with water (without particles) and air, for some seconds, in both directions (pulling and pushing with the syringe). Usually, a flux in the opposite direction w.r.t. the measurements was more effective to clean and unclog the channel. We avoided exposure of the mixtures to the air, to prevent contamination with dust. We tried to prepare diluted mixtures of particles and water not too long before their use, to avoid mold formation. Mixtures should be conserved in the dark, and should not be reused too much (not at all if possible). Water purification tablets and surfactant might help in preventing mold formation. We also paid attention to avoid contamination during dilution and preparation of the mixtures: we used different syringes, containers, etc...

- Clogging and particle clustering might disrupt the measurements. Microparticles are prone to stick together and to walls and surfaces. To reduce this unwanted effect, a small amount of surfactant (Triton X-100) was added to the mixtures. It was better dissolved in hot water. Even using surfactant, clogging was quite likely to happen if a high concentration of particles was used (we used 1/200 dilution w.r.t. an initial mixture with 5% volume content). To remove clogging, it was usually sufficient to strongly push the water in the opposite direction w.r.t. measurements. This was done also between each measurement, to avoid accumulation of particles in the circuit.

- The input mixture in the syringe was shaken at the start of each measurement session and every minute to ensure that the particles were homogeneously spread in the mixture (they tend to sink to the bottom). We kept the syringe almost vertical (around 30° tilted) pointing downwards during the measurements. We found that it was better to avoid that the input tube made U shapes, where the particles could accumulate because of gravity. Indeed, the fluid velocity in the tubes was much slower than in the channel, given the big difference in cross section. Whenever particles accumulated in a tube, they could get unstuck all at once and clog the microfluidic channel.

- Even if the syringe was actuated manually, it was possible to generate a sufficiently steady flow velocity. The average fluid velocity was obtained by measuring the time of syringe actuation and the volume of fluid injected. This was done at each measurement session, to provide a higher accuracy in the calculation of the mean velocity and an estimation of the variance. It was important to always leave an abundant quantity of air in the syringe, so that it acted as buffer. This helped to exert a sufficiently constant force to the syringe plunger.

- Training and testing the classifier using few samples (hundreds instead of thousands) and strong L2 regularization could provide a rough estimation of the goodness of scatterer configurations for the specific application, which can be useful in order to compare and choose among many configurations without performing extensive measurements.

4.5.1 Possible setup improvements

Even if we could perform satisfying measurements with the final version of the setup, there are still few possible improvements that we enlist here.

- Instead of actuating the syringe manually, it is more practical to employ a syringe pump (or an alternative fluid pump), which also ensures a much better control of the liquid flow. However, it is also necessary to mix the input mixture in the syringe, to avoid that the particles sink to the bottom. For example, this can be done by employing a mixing mechanism inside the syringe, such as a syringe magnetic stirrer.

- When several sessions of intertwined class measurements are performed, it becomes impractical to manually attach and detach the different syringes containing mixtures of particles from different classes and the flushing liquid. This problem can be overcome by employing switching valves, which allow to easily switch from one input tube to another. Furthermore, fully automated measurements could be possible, by interfacing programmable switching valves with the actuation of the pumping system.

- In order to further reduce the noise due to vibration and movement affecting the acquired patterns, the components that determine and influence the optical path could be mounted on a cage system mount.
- Instead of using a free-space laser, the noise could be decreased by conveying the laser light through an optical fiber with a suitable fiber collimator, which could be fixed to the other components. However, the benefits should be weighted against the sensitivity of optical fibers to thermal fluctuations and vibrational noise.
- Finally, a fast enough photodiode could be employed instead of an image sensor, in order to acquire the time evolution of the measured light due to passages of particles. Then, a classifier could be trained on the obtained time-dependent signals. Passing to single-pixel operations would remove the throughput limit posed by the camera frame rate, would eliminate the variability due to particle displacement along the flow direction, and would allow to detect and analyse every particle that passes through the channel.

4.6 Summary and conclusion

In this chapter we described the development of a simple label-free imaging flow cytometer to classify different sizes of transparent microspheres (similar in size to WBCs). We summarized step by step the investigations, the reasoning and the improvements that brought us to the final setup, measurements and machine learning approach, whose employment and results are discussed in the next chapter. We begun by describing a first realization of the experiment, where two synchronized cameras acquired the patterns projected by a flowing particle respectively with and without optical mixing performed by scatterers, which were implemented through an SLM. The main goal was to demonstrate an improvement in classification performance due to the scatterers, similarly as it was done by means of FDTD simulations in Chapter 3. Three undermining and correlated issues were detected: the classification was strongly affected by measurement bias, an insufficient SNR was achieved and bubbles were being generated in the fluidic circuit.

Therefore, the setup and the measurements were thoroughly analysed and many aspects were improved, such as the particle mixture preparation and employment, the background subtraction and detection, the prevention of bubble generation, the SNR and the measurement bias (mainly by the use of intertwined class measurements). After this, another version of the setup was considered, where the direct pattern (for particle presence detection) and the pattern optically mixed by the SLM were recorded by the same camera. Several scatterer configurations were explored and a complicated feature selection was employed in the attempt to achieve accurate and bias-free classification. However, we realized that the SLM flicker introduced too much noise in the acquired patterns, and that therefore we had to implement the scatterer layer in another way.

To this end, other physical scattering media were employed, such as PMMA microspheres sandwiched between two microscopic slides, diffusive media and double-axis diffraction gratings. Moreover, the setup was modified to acquire only one kind of pattern, since the enhanced SNR allowed to easily detect the presence of the particle directly from the pattern modified by scatterers. Further improvements to enhance the SNR were made and several scatterer configurations were explored. In order to facilitate this exploration, a significant effort was put in devising a way to approximately infer the performance of a scatterer configuration using only few hundred samples. In the end, the two most promising configurations were selected for the final realization of the experiment: one without scatterers (classification accuracy of around 94%) and one with one diffraction grating (classification accuracy of around 93%). Additionally, we listed the most important practical details that enabled us to reach a high enough quality of measurements and classification, together with some suggestions for further improvements.

The developed experiment was meant to provide a proof-of-concept of a novel classification method, based on the ELM paradigm, consisting in the application of a linear readout classifier directly on the pixel values of the acquired images. A strong focus was applied to the development of a suitable machine learning approach to achieve computationally cheap classification and to study and avoid the shortcut learning issues that undermined the experiment described in Chapter 2. We think that both of these are interesting aspects of a multidisciplinary topic that is rapidly expanding: particle classification requiring low computation can enable online operations in high-throughput imaging flow cytometers, and shortcut learning is a common but underestimated issue in machine learning applications to real-life problems.

References

[1] Emmanuel Gooskens. *Photonics and machine learning solutions for cell sorting.* European Master of Science in Photonics, 2018.

[2] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. *Shortcut learning in deep neural networks.* Nature Machine Intelligence, 2(11):665–673, 2020.

[3] Lei Yu and Huan Liu. *Efficient feature selection via analysis of relevance and redundancy.* The Journal of Machine Learning Research, 5:1205–1224, 2004.

[4] H. B. Mann and D. R. Whitney. *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other.* Ann. Math. Statist., 18(1):50–60, 03 1947.

[5] Ruey-Jen Yang, Lung-Ming Fu, and Hui-Hsiung Hou. *Review and perspectives on microfluidic flow cytometers.* Sensors and Actuators B: Chemical, 266:26–45, 2018.

5

Fast machine learning classification of microspheres in imaging label-free cytometry

In this chapter we discuss in detail the results obtained once the development of the setup, measurements and machine learning pipeline (described in the previous chapter) reached a satisfying level, guaranteeing accurate enough classification, reliable performance evaluations and fair comparison between scatterer configurations. This chapter represents an extension of the work published in [1].

Here we mainly consider the results obtained using the most promising scatterer configurations 1 and 4, with reference to Section 4.4 in the previous chapter. Even though the original goal of demonstrating a classification improvement due to scatterer layers was not achieved (further investigation is required), the experiment produced interesting results. In particular, these relate to the computational cost of the classification, the simplicity of the setup and of the machine learning implementation, and the methodology to address shortcut learning issues and ambiguity in comparing different configurations.

5.1 Interference pattern acquisition and machine learning classification

We acquired the interference patterns obtained by shining red laser light on transparent PMMA microparticles with diameters of (15.2 ± 0.5)µm (class A) and (18.6 ± 0.6)µm (class B). The particles were flowing in a 100 µm × 100 µm microfluidic channel (see Fig. 5.1 *a, b*). In particular, we refer to the configuration 1 described in Section 4.4, which will be referred to as NDG (from *no diffraction grating*). Additionally, we performed some measurements interposing a double-axis holographic diffraction grating between the microfluidic channel and the camera to modify the imaged pattern, which corresponds to configuration 4 in Section 4.4. This setup configuration will be referred to as DG (from *diffraction grating*) and the corresponding measurements and results will be discussed later on in Section 5.5. Figure 5.1 *c, d* show examples of the acquired background pattern for the two configurations.

It should be stressed that the particle class could not be straightforwardly determined by human examination, as can be noticed in Fig. 5.2, where we show some randomly selected examples of background-subtracted particle patterns belonging to the two classes. As expected, it can be noticed that well-centered particles (w.r.t. the illumination center) project concentric circular patterns on the camera, similarly as in the holograms analysed in Chapter 2 and as in the simulations based on Fresnel diffraction (previously performed by Emmanuel Gooskens). The acquired patterns characteristics were optimized by suitably adjust the distance between the microfluidic channel and the camera, so to maximize the classification accuracy. Not surprisingly, this resulted in patterns that are approximately centered w.r.t. the camera. Indeed, the considered linear classifier cannot easily generalize over different spatial displacements, as opposed to the more powerful, but also more computationally expensive, convolutional neural networks.

The classification inference process is schematized in Fig. 5.3. We performed background subtraction on each image by subtracting the previously acquired one. Because of our flow rates, the probability of having two consecutive frames containing significant particle signal is low. To ensure that the background subtraction did not introduce any significant artificial particle signal in the sample set, we discarded those images that directly followed an accepted one (image “acceptance” is described in the following lines). Since the CMOS sensor operated in a free-run mode, many of the acquired images contained the background illumination pattern without particles or with only a weak signal from particles far away from the illumination center. Instead of considering these unimportant images as an additional class for the machine learning classifier, we chose the simpler option of discarding them. This was done through the background

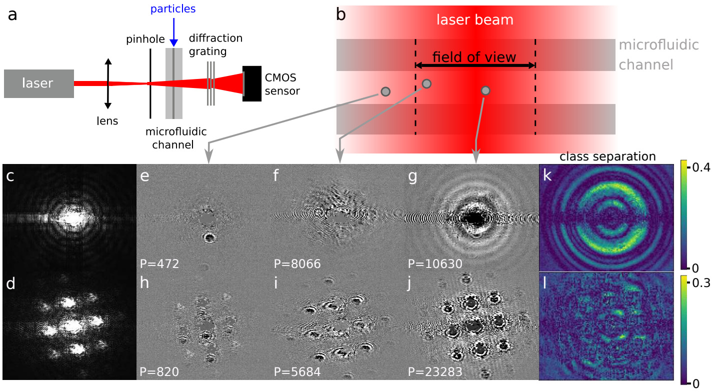


Figure 5.1: *a:* sketch of the employed setup. A PMMA microfluidic channel (cross section 100 µm × 100 µm) is illuminated by laser radiation (HeNe laser, $\lambda = 632.8$ nm) focused on a pinhole. The resulting beam passes through a transmissive double-axis diffraction grating (only in DG configuration) and is captured by a CMOS camera. *b*: drawing of the illuminated microfluidic channel region. The larger the particle distance from the field-of-view center, the weaker the acquired particle signal (measured by the perturbation quantity P). *c-l*: respectively for the NDG (top row) and DG (bottom row) configurations, examples of background pattern (1^{st} column), background-subtracted particle patterns with increasing intensity (2^{nd} to 4^{th} columns) and class separation colormaps (last column). *e* and *h* are well below the respective acceptance thresholds, in this case $\theta_P^{NDG} \approx 7200$ and $\theta_P^{DG} \approx 5100$ (for a particle ratio $R = 0.04$). *f* and *i* are just above and *g* and *j* are well above the respective acceptance thresholds. Grey arrows suggest a qualitative link between these examples and the particle position w.r.t. the field-of-view shown in *b*.

detection algorithm: to measure the strength of the particle signal, for each background-subtracted pattern we calculated the sum of all the squared pixel values, which from now on will be referred to as *overall perturbation* P. Examples of background-subtracted images with the respective P values are shown in Fig. 5.1 *e, f, g* and *h, i, j* respectively for the NDG and for the DG configurations. Only those images whose P value is larger than a chosen *acceptance threshold* θ_P were accepted as samples used to train and test the machine learning classification. The criteria and the motivation for the choice of θ_P will be explained in detail later on in this section.

Similarly as in Chapter 3, for this experiment we trained and tested a simple linear classifier based on *logistic regression* (see Chapter 1), directly applied to the pixel values of background-subtracted images. Its task was to classify the ac-

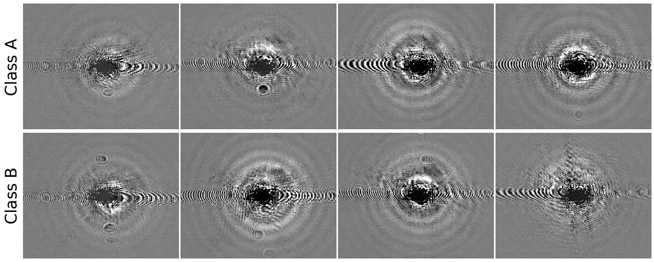


Figure 5.2: Examples of particle patterns (after background subtraction) acquired with the NDG configuration, randomly selected from class A measurements (*first row*) and class B measurements (*second row*). It can be noticed that the classification is not straightforward (maybe impossible) by human eye.

quired interference patterns according to the microbead diameter. We employed *L2 regularization* to reduce overfitting and we optimized its strength by means of k-fold cross-validation, with number of folds $N_s = 11$ (see next section for details). When images with high resolutions ($> 10,000$ pixels) were employed as classification samples, a feature selection procedure was applied to reduce both the risk of overfitting and the training time. In particular, we discarded those pixels that showed low *class separation* (see Section 4.4 in the previous chapter), i.e. where the value distributions corresponding to the considered classes showed a small difference. The calculated class separation can also provide interesting insight on which areas of the acquired interference pattern are most relevant to the classification purpose (Fig. 5.1 *k, l*).

Measurement details

The employed PMMA microbeads mixtures were obtained by diluting the original mixtures (5% solid content volume) in a solution of water and a small quantity of surfactant and a water purification tablet, reaching a fraction of solid content volume of 0.024%. The mixtures were pumped in a $100\,\mu\text{m} \times 100\,\mu\text{m}$ straight PMMA microfluidic channel at a constant rate of ≈ 0.003 ml/s, using three different syringes (one at a time) respectively for the two particle classes and the flushing water, to avoid particle contamination. Between each measurement session, the microfluidic channel and tubes were flushed with water to remove possible residual microbeads.

The microfluidic channel was illuminated by focusing HeNe laser radiation (constant emitted power of 3.5 mW) on a pinhole (diameter of $25\,\mu\text{m}$) tightly clamped to the microfluidic slide in order to prevent it from moving during

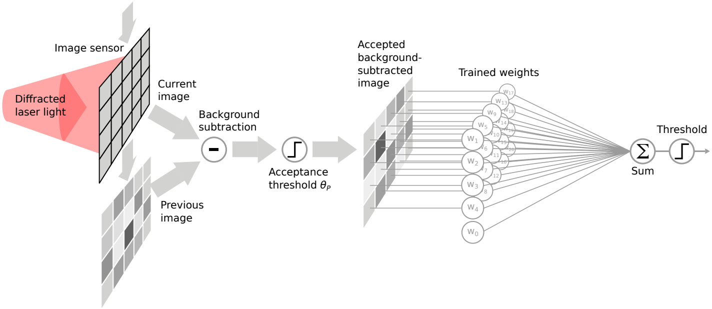


Figure 5.3: Sketch of the machine-learning classification pipeline. Intensity patterns are acquired by the image sensor in free-run mode. The difference between consecutive images is calculated (background subtraction), and if the squared sum of its pixels is lower than a chosen acceptance threshold value θ_P the image is considered as background and discarded. A linear classifier (trainable weighted sum) is applied to accepted background-subtracted images. If the outcome is positive, the analyzed particle is classified as belonging to class A, to class B otherwise.

measurements and to reduce vibration noise. When employed, the holographic diffraction grating film was directly attached to the front side of the microfluidic slide. Images of 632×508 pixels were acquired in free-run mode by a Ximea MQ013MG-0N camera, at a frame rate of around 138 fps and with 29 µs exposure time.

Machine learning details

The whole image processing presented in this work was executed in Python. In particular, the machine learning pipeline was built on top of the *scikit-learn* library [2] and the following functions were employed: *model_selection.GroupKFold* to implement the two nested cross-validation loops; *preprocessing.StandardScaler* to normalize the features before each training or inference step; *linear_model.LogisticRegression* with “l2” penalty, “liblinear” solver and “balanced” class weight, as linear classifier. The only optimized hyperparameter was the inverse of the L2 regularization strength C, chosen among 13 values equidistant in log. scale from 10^{-5} to 10. The downsampling to desired image resolutions was performed employing the “block_reduce” function from the *Scikit-image* Python library. The classification error rate reported in the box plots represents the fraction of misclassified test samples w.r.t. the total number of test samples, thus it is the complementary percentage of the classification accuracy.

5.2 Effects and prevention of measurement bias

In this Section, we take a closer look at the problem of measurement bias, which was encountered and discussed also in Chapter 2 and in the previous chapter. We present results regarding dedicated measurements designed to show the effects of measurement bias in the experiment. Moreover, we describe the approach we employed to treat the problem, which is based on intertwined class measurements and a suitable validation algorithm, and we demonstrate its effectiveness.

Let us first review the measurement bias problem. Supervised machine learning algorithms can learn how to carry out a certain task on a given sample population, e.g. classification of cells in digital pictures, by analyzing a set of training samples for which the solution to the task (i.e., the training label) is given. Therefore, the performance of such algorithms when applied on unseen samples (generalization) is obviously limited by how comprehensively the training samples represent the target sample population. When the noise in the samples and the labels are uncorrelated, generalization can be usually improved by increasing the number of training samples or through regularization techniques (see Chapter 1), i.e. reducing the overfitting. This is a very well-known practice and in this case the presence of overfitting can be easily detected by testing the algorithm on samples that were not used in the training stage (e.g. through cross-validation). Less known and more deceptive is the case where the noise and the training labels are correlated, e.g. in classification problems where samples from different classes are acquired or measured under significantly different experimental conditions. In this case, which we refer to as measurement bias, the machine learning training is most likely biased by the measurement conditions, which are mistakenly considered as a distinguishing trait of the classes. This leads to a worsening of the classification performance under new measurement conditions, i.e. to a decrease in generalization. The elusiveness lies in the fact that measurement bias leads to misleadingly high estimated accuracies and cannot be detected if the training and test samples are measured under the same biasing conditions.

To apply this more concretely to the case of an imaging microflow cytometer, e.g. to train a label-free white blood cell classifier, for practical reasons, monocytes and granulocytes might be kept separated and the samples might be acquired in different measurement sessions, often leading to measurement bias because of drift inbetween sessions. Indeed, this case many factors may produce significant drift in measurement parameters, such as fluctuations of the light source properties, displacement or distortion of the optical beam (e.g. due to thermal expansion of some elements), refractive index changes of the optical components (e.g. due to slow water absorption of the microfluidic channel walls) and so on. As we have shown in Chapter 2 and as we demonstrate in this Section, in this case background subtraction might mitigate but cannot completely remove the measurement bias. Indeed, the background signal is given by the unperturbed laser beam impinging on the camera screen while the particle signal is mainly given by a spatial optical path perturbation of the same laser beam. These two signals are combined in a strongly nonlinear way by the image sensor measurement and therefore they cannot be decoupled by a simple linear operation such as background subtraction.

Another approach to remove measurement bias is to mix the two types of cells and determine their class (i.e. their label) during the image acquisition using an auxiliary system, e.g. a fluorescent label detector. However, including such a system is more complex, also considering that to train an accurate classifier even more accurate ground truth data is required.

In order to provide a direct experimental demonstration of the negative effects of measurement bias, we performed ad-hoc sample measurements according to the following chronology:

$$A_{\text{train}}(20 \text{ mins}),\ B_{\text{train}}(20 \text{ mins}),\ A_{\text{test}}(2 \text{ mins, } 15 \text{ s}),\ B_{\text{test}}(2 \text{ mins, } 26 \text{ s}) \quad (5.1)$$

Here A and B refer to interference patterns acquisition of PMMA beads with diameter of 15.2 µm and 18.6 µm respectively. Even employing a proper cross-validation technique, using samples from A_{train} and B_{train} for training, validation and test, the employed particle classification provides on average significantly lower test errors than when A_{test} and B_{test} are employed for testing (Fig. 5.4, compare *left* with *middle*). This means that the classifier training was influenced by the measurement conditions leading to an overestimated generalization capability when samples from the same measurement session were employed for testing. Such an effect is also responsible for a large variance in performance evaluation ascribed to the fluctuations of the measurement conditions during the measurement sessions.

In this work, we developed a simple method to solve this problem, i.e. to effectively decouple the training sample labels from slow fluctuations of the measurement parameters, avoiding measurement bias. In particular, we acquired the samples according to the following measurement sessions chronology (duration of 2 mins each):

$$A_1,\ B_1,\ A_2,\ B_2,\ \ldots,A_{N_s},B_{N_s} \quad (5.2)$$

i.e. using intertwined class measurements to provide training, validation and test samples to the classification algorithm. In all cases, the measurement sessions were performed at different times in the same day. Considering a number of

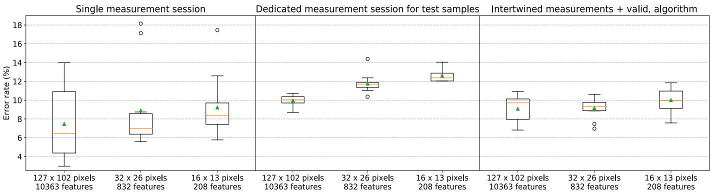


Figure 5.4: Box plots of the classification error evaluated by means of cross-validation on images down-sampled to different resolutions (x axis). Each box represents the distribution of the N_s error values, corresponding to different folds, obtained through k-fold cross-validation. Boxes, whiskers, orange lines and green triangles respectively represent the interquartile range, the range, the median and the mean of the error values. The outliers (outer points distant more than $1.5\times$(interquartile range) from the interquartile range) are represented by circles. The employed samples were selected among the acquired images considering a particle ratio value of $R = 0.04$ (see Section 5.3). *Left*: the samples employed for training, validation and test were obtained from a single measurement session per class, providing misleadingly low average errors and high variance due to measurement bias. *Middle*: test errors are evaluated on samples from dedicated measurement sessions, showing the correct generalization capability of the trained classifier. *Right*: the proposed intertwined class measurements and validation algorithm were employed to remove the measurement bias influence from classification training, validation and test. The comparison with the middle box plot shows an improved generalization capability of the trained classifier.

sessions per class $N_s = 11$, we then employed the following *validation algorithm*:

```
for i = 1,2, ..., $N_s$:
    $A^i_{\text{train}} \leftarrow \bigcup_{n \neq i} A_n \quad , \quad B^i_{\text{train}} \leftarrow \bigcup_{n \neq i} B_n$
    for j = 1,2, ..., $N_s - 1$:
        $A^{ij}_{\text{train}} \leftarrow \bigcup_{n \neq i,j} A_n \quad , \quad B^{ij}_{\text{train}} \leftarrow \bigcup_{n \neq i,j} B_n$
        for k = 1,2, ..., $N_h$:                                        (5.3)
            $\theta_{ijk} \leftarrow \text{train classifier}(A^{ij}_{\text{train}},\ B^{ij}_{\text{train}} \mid h_k)$
            $p_{ijk} \leftarrow \text{test classifier}(A_j,\ B_j \mid \theta_{ijk},\ h_k)$
    $\tilde{h}_i \leftarrow \text{select best hyperparameter}(p_{ijk})$
    $\theta_i \leftarrow \text{train classifier}(A^i_{\text{train}},\ B^i_{\text{train}} \mid \tilde{h}_i)$
    $p_i \leftarrow \text{test classifier}(A_i,\ B_i \mid \theta_i,\ \tilde{h}_i)$
$p_{\text{final}} \leftarrow \text{average}(p_i)$
```

where h_k is a hyperparameter (L2 regularization strength in our case) to optimize by choosing among given options corresponding to $k = 1, 2, ..., N_h$ and $\tilde{h}$ is the chosen hyperparameter value; θ is the set of readout parameters (weights and intercept) determined by the training, p refers to a performance evaluation (the estimated accuracy in this case) of the machine learning classifier and p_{final} is the final evaluation of the whole algorithm, including the hyperparameter selection. The generalization of the algorithm to multiclass and multiple hyperparameters cases is straightforward.

The main concept here is that the training, validation and test datasets not only are always disjoint as it happens in traditional cross-validation, but they were also acquired in different and chronologically separated measurement sessions. Even though it should be considered good practice, this kind of methodology is often not implemented [3] and in this work we show some possible misleading consequences.

Applying the proposed intertwined measurements and validation algorithm, we obtained better classification performance (Fig. 5.4, compare *right* with *middle*). Moreover, we obtained an evaluation of the accuracy average and variance generalized to different measurement sessions. As explained in the next section, we checked if the measurement bias was still affecting our results by means of a suitable test. The number of sessions per class N_s should be chosen high enough to ensure that the measurement bias is removed and to achieve a satisfactory generalization capability of the trained classifier. Generally, N_s is limited by the difficulty and the time required to perform a high number of measurement sessions to provide training samples. Therefore, an optimal N_s is highly application-dependent.

5.3 Classification performance vs. field-of-view

Depending on how displaced along the channel the flowing particle is w.r.t. the laser beam center, the acquired interference patterns may vary in intensity, position and shape (e.g. Fig. 5.1, *c, e, f*), making the particle analysis more or less difficult. The range of such a displacement for which it is still possible to perform the particle classification is called *field-of-view* (FoV) of the cytometer (see Fig. 5.1 b). In our case, the time interval between two consecutive image acquisitions is much longer than the travel time of a particle through the FoV, implying that a fraction of the flowing particles are not measured. Thus, the larger the FoV, the higher the number of particles that are analysed w.r.t. the total number of flowing particles and therefore the higher the maximum *sensitivity* of the cytometer. Usually the sensitivity of particle detection can also be enhanced by employing an effective microfluidic focusing system [4] (to reduce transverse displacement), even though there is a trade-off between fabrication complexity, sensitivity and throughput. In any case, the particle displacement along a microfluidic channel always constitutes an important source of variability.

In this experiment, we estimated the classification performance considering different one-dimensional FoV values along the microfluidic channel direction. The transverse channel dimensions were neglected since the illumination was considered to be relatively uniform over the channel cross section. As it is intuitively schematized in Fig. 5.1 *b*, the larger the distance of a particle from the illumination center, the smaller the P value of the obtained image. This implies that the FoV is determined by the choice of the acceptance threshold θ_P. Still, two particles belonging to different classes and in the same position will lead to two images with different P value. Therefore, in order to have the same FoV for different classes of particles, the applied θ_P should ideally be class-dependent. However, this is only feasible in the training stage, where the classes (labels) are known, while in the test stage a common acceptance threshold has to be used for all the acquired images. To avoid a mismatch between training and test sample populations which may be detrimental for classification performances, in this work we chose to use a common θ_P for the two classes in both training and testing. In practice, the applied acceptance threshold θ_P was chosen so that a desired value for the *particle ratio* R, defined as the ratio of the number of accepted particle images to the total number of acquired images, is obtained. The reason is that the particle ratio can be used as a more objective bridge quantity in the classification comparison with the cases where diffractive optical layers are interposed between the microfluidic channel and the camera (this is explained in Section 5.5). For each value of R, the FoV for each class can be estimated (see appendix Section 5.8).

We evaluated the performance of the presented classification algorithm considering sample sets obtained through different choices of $R = 0.02,\ 0.04,\ 0.06,\ 0.08$ (see Fig. 5.5). The employed image resolution is 127 x 102 pixels, corresponding to a down-sampling with a factor 5 w.r.t. the camera resolution. A feature selection algorithm (which used the class separation measure for scoring) was applied to remove the most noisy pixels and therefore to decrease the risk of overfitting, leaving a total of 10,363 features, i.e. around 80% of the pixels.

The sample set corresponding to $R = 0.04$ provides the best classification performance (low error average and variance) due to a trade-off between the quality and the number of samples. Indeed, a lower R, or equivalently a higher acceptance threshold θ_P, means we only keep the samples with the highest quality in the center of the laser beam, reducing the FoV. This results in a lower sample variability (which should make the classification easier), but also in a lower number of available samples (which makes it more difficult to train the classifier). It should be stressed that the optimal R value is application-specific. In particular, R should be chosen so that the classification accuracy is maximized, while trying to achieve the target cytometer throughput. Moreover, the number

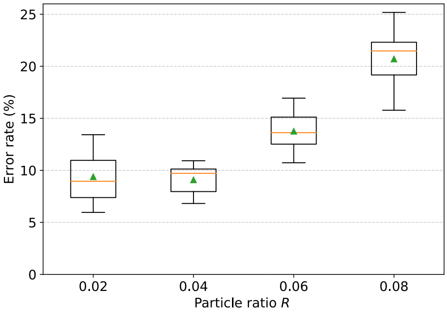


Figure 5.5: Box plot of the classification error evaluated by the proposed validation algorithm on sample sets obtained through different choices of R (on the x axis), i.e. applying different acceptance thresholds. $R = 0.04$ provides the best classification performance (low error average and variance) due to a trade-off between the field-of-view and the number of samples N.

of available training samples and the classifier complexity (e.g. given by the image resolution) both play a major role in the choice of R, because of the need to avoid overfitting.

We furthermore double-checked whether the classifier would still be biased by the measurement conditions, in spite of our intertwined class measurements. This was done by training it on the same dataset but with half of the measurement sessions mislabeled, i.e. in list (5.2): $A_2 \to B_2$, $B_2 \to A_2$, $A_4 \to B_4$, $B_4 \to A_4$, and so on. In this way, the characteristic features given by the different sizes of the beads (corresponding to the true classes) were equally present in both of the nominal classes (those presented to the training algorithm). Thus, if the classifier only learns the particle-related features and therefore is not biased, it would provide the same accuracy of a random guess ($\approx 50\%$ in the two-classes case). This *uniform mislabelling* (UM) test shows indeed errors around 50% in Fig. 5.6, which indicates that no significant bias is detected. This result demonstrates that our intertwined class approach is effective in removing measurement bias.

5.4 Classification performance and time vs. image resolution

In imaging flow cytometry, the resolution of the acquired images is a key parameter, not only because of the obvious relation with the price and the frame rate of the employed image sensor, but also because it greatly influences the execution

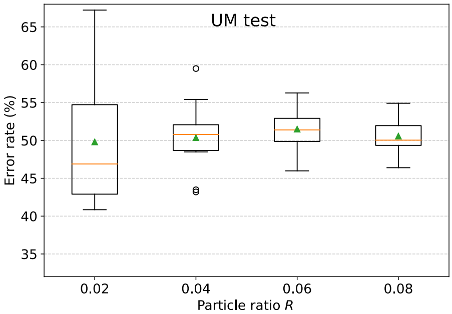


Figure 5.6: Box plot of the classification error evaluated through the proposed UM test, corresponding to the classifications results presented in Fig. 5.5. The training is performed on uniformly mislabelled data and therefore the obtained test error is expected to be $\approx 50\%$ (random choice) for our two classes, if the learning is not affected by measurement bias.

time of the particle analysis/classification and therefore the throughput limit of online operations, such as cell sorting.

We evaluated the performance of our particle classification technique for different resolutions of the employed images and we estimated the corresponding execution (inference) times. Different sample sets were obtained by downsampling the acquired images by approximately 2, 5, 10, 20, 40, 100 and 400. Thus, the original resolution of 632×508 pixels was decreased respectively to 316×254, 127×102, 64×51, 32×26, 16×13, 7×6 and 2×2. Note that for the highest two resolutions respectively 87.1% and 20% of the downsampled pixels were discarded by means of the feature selection, in order to limit overfitting and the computational cost of training the classifier. For the remaining resolutions, no feature selection was performed, i.e. all the pixel values were employed as features for machine learning.

Using the previously determined optimal particle ratio value $R = 0.04$, we obtained classification errors below 10% for image resolutions of 127×102, 64×51 and 32×26 pixels (Fig. 5.7 *a*). The error is just slightly worse using 16×13 pixels, but it abruptly increases for 7×6 and 2×2 pixels, showing that the resolution is too low to provide the classifier with enough particle information. In particular, this shows that the classification task could not be carried out by just considering the total forward scattering intensity, as in bead size discrimination in traditional flow cytometers. This suggests that our classification system presents much less stringent requirements on the alignment of flowing particles with the laser beam. Indeed, both classes contain particle patterns with various overall perturbation values, whose ranges overlap for the most part. Also because of this, the classification could not be successfully carried out by just considering the overall intensity of the considered patterns. The box plot also shows that selecting 12.9% of the pixels from highest resolution images (316×254 pixels) leads to a small but significant degradation of the classification performance. Setting $R = 0.02,\ 0.06$ or 0.08, similar performance trends with an overall degradation were obtained. It should be stressed that the relation between the classification error and the image resolution depends on the addressed classification task and cannot be generalized.

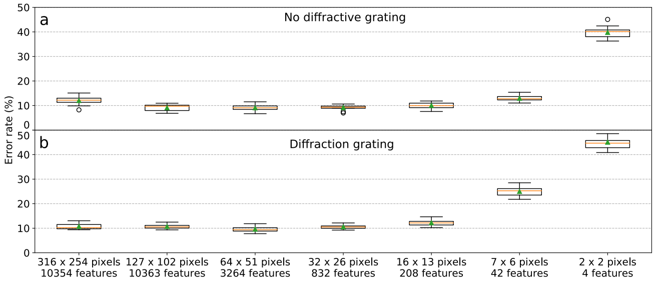


Figure 5.7: Box plots of the classification error for $R = 0.04$ evaluated on particle images of different resolution (x axis), respectively with (*a*) and without (*b*) holographic double-axis diffraction grating interposed between the camera and the microfluidic channel. Classification errors lower than 10% were obtained for image resolutions down to just 32×26 pixels. Generally, the interference patterns processed by the diffraction grating provide particle classification with similar or slightly higher errors. Note that the number of features used to evaluate the first two points was further reduced by feature selection.

The average inference execution time of the classification algorithm (i.e. background subtraction + application of acceptance threshold + machine learning inference, see Fig. 5.3), was evaluated for different image resolutions running a Python script on a normal laptop (Intel Core i5-8250U, 1.60GHz × 8). Ultrafast image classification was achieved with computational times per particle in the order of 100 µs to 10 µs depending on the resolution (Table 5.1). It should be stressed that these values could be easily further decreased by, e.g., employing multi-core computing, a graphics processing unit (GPU) or dedicated hardware.

We achieved considerably low execution times because we could skip the most complex and computationally expensive operations that are usually performed in digital holographic microscopy and image classification. These are the image reconstruction from the acquired hologram and the feature extraction, which usually consists in calculating suitable mathematical representations that simplify the work of the readout linear classifier. The calculated features can be engineered or optimized separately from the training algorithm, or can be treated as hyperparameters or even completely integrated in the neural network training and structure, such as in convolutional NNs. In any case, feature extraction generally implies that nonlinear calculations are added to the classification inference operations, significantly increasing the corresponding computational cost. In our case, a further decrease in computational cost could be possible by skipping the background subtraction. However, this would require a strong enhancement of the SNR, e.g. by employing a more powerful laser and a smaller pinhole.

Image resolution (pixels)	316×254	127×102	64×51	32×26	16×13	7×6	2×2
Classification time (μs)	200	38	19	13	10	9.0	8.8

Table 5.1: Execution time per particle of the proposed classification algorithm for different image resolutions, evaluated on a laptop (Intel Core i5-8250U, 1.60GHz × 8) using a Python script (Numpy library). The reported time values are averaged (median) over 10,000 iterations of the following steps: computation of the difference between the target and the background image after conversion to float type matrices; application of the acceptance threshold to the sum of the squared elements of the difference matrix; weighted sum of the difference matrix (i.e. machine learning inference).

5.5 Classification employing diffractive layers

From a machine-learning perspective, one might intuitively assume that applying a simple linear classifier on the raw pixel values of an image would generally provide a much weaker classification power w.r.t. common approaches based on feature extraction and deep learning. Nevertheless, as we previously discussed, linear classifiers and regressors can provide state-of-the-art performance when applied to random high-dimensional nonlinear transformations of the input, as it happens in widespread approaches like *Extreme Learning Machines* [5, 6] (ELM) and *Reservoir Computing* [7, 8] (RC). Indeed, the relation between the optical particle features and the detected interference pattern (input and output) is mathematically nonlinear and the high number of pixels in an image sensor can potentially provide a high-dimensional mapping. Therefore, modulating and controlling the interference pattern projection e.g. through interposed diffraction layers can provide an extremely fast and power-efficient source of computational power, as it was experimentally demonstrated in [9, 10]. Moreover, in Chapter 3 we numerically demonstrated that random diffractive layers that resemble diffraction grating structures can significantly improve the performance of a linear classifier in non-trivial classification of cell structures.

However, by interposing diffractive layers between the particle and the image sensor, the automatic discrimination of particle images from background images is likely to be influenced. In particular, it can modify the cytometer sensitivity and the class balance in the training sample sets. In this section we present a method to avoid these issues and to guarantee a valid performance comparison, laying the groundwork for hardware-based improvement of the proposed classification technique.

Here we present the comparison between the two considered configurations (NDG and DG). In practice, in order to have a fair comparison, the main issue is how to choose the corresponding acceptance thresholds θ_P^{NDG} and θ_P^{DG} to make the two cases comparable. We want to compare both cases for a fixed maximum sensitivity of the cytometer, i.e. when the FoV is the same in both configurations. Generally, the introduction of a diffractive layer changes the intensity of the acquired particle signal in a nonlinear way, so that $\theta_P^{NDG} = \theta_P^{DG}$ or even $\theta_P^{NDG} \propto \theta_P^{DG}$ would lead to different FoVs. However, as we will discuss in the calculation of the field-of-view in appendix Section 5.8, there is a one-to-one correspondence between the *particle flow rate* R_f, the particle ratio R and the field-of-view. If we can guarantee in our experiments that the particle flow rate R_f is constant, the requirement of having a fixed field-of-view translates to a requirement of having a fixed particle ratio R. This allows us to set the acceptance thresholds for both configurations, by looking at the experimentally determined relationship between the particle ratio R and the acceptance threshold θ_P (see Fig. 5.8 left plot).

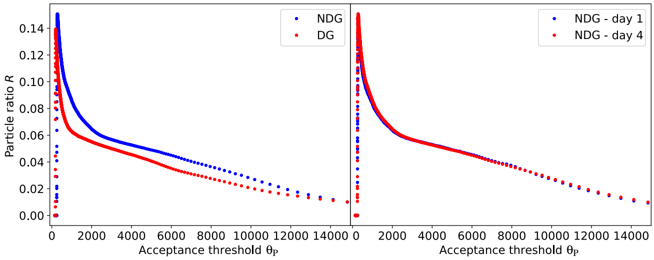


Figure 5.8: Particle rate R as a function of the acceptance threshold θ_P for different measurement sessions. *Left*: comparison between the configuration without interposed diffraction grating (NDG, blue dots) and with diffraction grating (DG, red dots). The diffraction grating changes the relation in a nonlinear way. *Right*: comparison between measurement sessions (both in NDG configuration) performed with a time separation of 3 days. The curves do not change significantly from one measurement session to another, indicating stability in our measurements.

We also checked whether the particle flow rate R_f did not change significantly from one measurement session to another, and therefore that the relation between R and θ_P remained constant. This was experimentally confirmed by comparing two measurements in NDG configuration performed at significantly distant times (3 days one from another, see Fig. 5.8 right plot).

Generally, the DG configuration provided similar or (in most cases slightly) inferior classification performance compared to the NDG configuration (for example compare Fig. 5.7 *a* and *b*). The fact that we could not demonstrate an advantage in classification performance thanks to the use of scatterer layers has different possible explanations. Here we list the ones that we believe to be the most probable.

- The possibility of classification improvement using scatterers could have been offset by the significantly lower SNR, caused by the intensity attenuation due to the diffractive layer.
- It is also possible that the nonlinear mapping performed by the image detection (where nonlinearity is given by the conversion from optical to electric domain and pixel saturation) was already close to optimal in the NDG case, for the specific task and readout classifier. In fact, the main challenge of the considered classification task is the variability due to the microbead displacement w.r.t. the illumination center, which can be in principle arbitrarily alleviated by decreasing the cytometer FoV. In that case, we expect that a properly designed diffractive layer may improve the classification performance, especially when the particle types are distinguished by differences in internal structure, such as in sorting of white blood cells, as indicated by the simulations in Chapter 3.
- Or simply, a more extensive or thorough exploration of different scatterer layers and configuration was needed to find a more favorable setting.

However, even without demonstrating advantages due to scatterers, the obtained classification results are still interesting, since we showed that particle classification with a large FoV w.r.t. particle diameter is possible with a very simple and cheap setup, at an extremely low computational cost. A comparison with other relevant works is discussed in the next section. It should be also stressed that these results would have not be achieved without a proper analysis and treatment of measurement bias, which we could not find in other literature about label-free imaging flow cytometry.

On the other hand, the fact that the classification performance is not significantly disrupted by the heavy deformation of the particle interference pattern due to the diffraction grating (visual examples are in Fig. 5.1 *d, h, i, j, l*), demonstrates the robustness of the proposed cytometry implementation. Indeed, the classifier can be trained without any problem when the acquired images are altered, e.g. by fabrication defects, misalignment or blurring (see also Fig. 4.19, as long as the particle information regarding the difference between classes is not lost. Such a versatility can represent a significant advantage w.r.t. other machine learning implementations where priors (see Chapter 1) based on human visualization are employed. This is relevant in practice, as motion blur is a common problem in imaging flow cytometry [4] and it often limits the achievable throughput.

5.6 Comparison with other works

In this section we compare the classification performance of our method with the performance presented in other three comparable works, reporting online label-free classification (Table 5.2). It should be specified that the throughput of our setup is quite low (around 2.7 classified cells per second for $R = 0.04$), since our work mainly focuses on general machine learning aspects of label-free imaging flow cytometry rather than on developing the flow hardware that would enable a high-throughput device. We should also stress that it is difficult to estimate and compare the complexity of the respective classification tasks, since not only do the particle characteristics play a crucial role, but also cytometer properties such as the FoV, the presence of an image focusing system or the control of measurement bias.

Classification task	Classifier	Image resolution	Imaging method	Image FoV	Classification performance	Accel.	Ex. time / particle	Meas. bias control
Beads with diameters of 7, 10 and 15 µm [11]	CNN	21×21	Microscope	Centered, cropped	93.3% mAP	GPU	< 1 ms	Unreported
3 white blood cell (WBC) types [12]	Rand. forest on extracted features	31×31	Lens-free - raw hologram	Unreported	96.8% accuracy	GPU	0.2 ms	Unreported
A WBC type and an epithelial cancer cell [13]	Deep CNN	Unreported	Time-stretch microscope	25 µm along flow	95.74% accuracy	GPU	3.6 ms	Unreported
Beads with diameters of 15.2 and 18.6 µm (our work)	Linear (log. regression)	32×26	Lens-free - raw hologram	≈ 300 µm along flow	> 90% accuracy	None	0.013 ms	Yes

Table 5.2: Comparison of machine learning-related aspects regarding three other works (reporting online label-free classification via particle imaging) and our work. CNN is the acronym for *Convolutional Neural Network*, while mAP is the abbreviation of *mean Average Precision*.

In particular, it should be stressed that a wider FoV not only introduces the challenge of generalizing the classification to a higher variability in particle position, but also implies a smaller contrast of the particle signal w.r.t. the background illumination. In this regard, in [13] the reported FoV is 25 µm, much smaller than what we estimated for this work (≈ 0.3 mm, see Table 5.3). While in [12] there seems to be no mention of it, in [11] the FoV is comparable with ours, but the actual machine learning classification is applied on cropped and centered particle images so that the variability in particle position does not complicate the classification. Furthermore, it is interesting to note that our classification algorithm is not specifically built to extract position-invariant features, as opposed to the classifiers used in the other works here described. Finally, a distinguishing trait of this work is that the classifier could learn and operate on images that could not be straightforwardly classified or recognized by human inspection (e.g. see patterns in Fig. 5.2).

This said, the presented bias-free classification is at least 15 times faster w.r.t. the aforementioned works, even if it is only computed with a common laptop and without GPU acceleration.

5.7 Summary and conclusion

In this chapter we discussed some important machine-learning aspects regarding fast particle classification with label-free imaging flow cytometry. The development of the considered experiment was described in the previous chapter. In Section 5.2 we demonstrated that we could detect and prevent measurement bias employing intertwined class measurements and a suitable validation algorithm. Then, in Section 5.3, we studied the relation between the obtained classification performance and the field-of-view of the cytometer. In Section 5.4 we presented the accuracy and the execution time of the classification inference for different resolutions of the employed images. Furthermore, in Section 5.5 we discussed a method to properly compare the classification results using different scatterer configurations, by enforcing the constraint that the field-of-view should be the same. Finally, in Section 5.6, we put the obtained results in context by comparing our experiment with other three related works.

We employed a simple, cheap and compact cytometer and demonstrated the classification of particle interference patterns at a very low computational cost, which for example can enable online high-throughput analysis (e.g. for cell sorting). Proof-of-principle experiments were performed by acquiring and classifying interference patterns projected by transparent PMMA microparticles with diameters of (15.2 ± 0.5)µm and (18.6 ± 0.6)µm, that could not be easily classified by human inspection. In particular, we discussed and demonstrated the following fundamental aspects:

- Detection and treatment of a deceptive kind of shortcut learning (measurement bias) that can affect machine learning models in the field, rising from the correlation between the ground truth information (necessary for training and testing) and the experimental conditions that may influence the measurements.

- Direct application of a linear classifier on background-subtracted images of particle interference patterns, allowing simple and robust machine learning classification of particles with high position variability (the FoV is much larger than the particle size) at an extremely low computational cost.

- A method to properly evaluate the change in classification performance when a diffractive layer (a double-axis holographic diffraction grating film in this case) is interposed between the camera and the microfluidic channel, making sure that the field-of-view (i.e. the sensitivity) and the class balance of the training sample sets remain unchanged.

As we discussed in Chapter 3, a diffraction layer interposed between the camera and the microfluidic channel can in principle improve particle classification according to the Extreme Learning Machine (ELM) paradigm, even though in this case similar or slightly worse performance was achieved. Nevertheless, we think that an experimental demonstration of the classification improvement due to an interposed diffractive layer should be tried via a more extensive exploration of different configurations and/or considering a more morphology-based classification task, such as in white blood cell sorting.

Quantitatively speaking, the best achieved performance in terms of classification accuracy and execution time are an accuracy above 90% (on 32×26 pixels images) with an estimated execution time of 13 µs (using a common laptop) and a field of view of around 300 µm along the microfluidic channel. We believe that the accuracy can be further enhanced by simply employing a smaller field-of-view and by acquiring a sufficient number of samples to properly train the classifier. As mentioned, suitable measurements, validation algorithms and tests were devised and employed to obtain a correct training and evaluation of the classification performance, which would otherwise have been biased by slight drifts of the measurement conditions. The proposed particle classification algorithm is at least one order of magnitude faster w.r.t. the state-of-the-art, represented by other three works regarding fast online classification in label-free flow cytometry [11–13], where instead GPU acceleration was employed.

The low computational cost of the proposed classification method could enable ultrafast (around 100,000 particles/s) online particle analysis if applied to optofluidic time-stretch microscopy [14, 15], removing or alleviating the issue of storing large amounts of data and allowing fast online operations in these systems, such as cell sorting. Another possible high-throughput application is to perform the cell analysis in parallel employing multiple particle streams, where the computational cost would be a bottleneck parameter [16, 17].

Finally, the all-round simplicity and the low cost of the presented flow cytometry approach make it suitable for compact point-of-care applications, where both the training and the use of the cytometer should not require high technical expertise.

5.8 Appendix: calculation of acceptance threshold and field of view given a chosen particle ratio

The relation between particle ratio R and acceptance threshold θ_P was graphically obtained by plotting the count of accepted images divided by the total number of images for many values of θ_P (Fig. 5.8 left plot). It was then straightforward to select an acceptance threshold corresponding to a chosen particle ratio.

The field of view (FoV) can be derived from the acceptance threshold, knowing the aforementioned *particle flow rate* R_f, the *exposure time* τ and the *fluid velocity* v. In particular, let us start by finding the probability that an image contains enough particle information, i.e. that a particle is at least partially present in a given FoV during an exposure time interval τ. Let us call t_{in} and t_{out} the times at which a particle respectively enters and exits the FoV. Then, let us call τ_{start} and τ_{end} the start and end times of the camera exposure. Thus, the conditions for capturing the signal of a particle in the FoV are $t_{in} < \tau_{end}$ and $t_{out} > \tau_{start}$. We can substitute $t_{out} = t_{in} + \text{FoV}/v$, being FoV/v the time that a particle takes to travel through the FoV, obtaining $\tau_{start} - \text{FoV}/v < t_{in} < \tau_{end}$. Since the density of particles in the mixture is quite low, we can consider the passage of particles as independent events. Therefore, the process of imaging the pattern from k particles in the FoV can be considered as the Poisson process describing the occurrence of k events t_{in}, with a time rate R_f, in a time interval $\tau + \text{FoV}/v$, with probability:

$$Pr(k, \tau + \text{FoV}/v, R_f) = \frac{[R_f(\tau + \text{FoV}/v)]^k}{k!} e^{-R_f(\tau+\text{FoV}/v)} \tag{5.4}$$

In our case $\tau = 29\,\mu\text{s}$ and we can calculate R_f by multiplying the flux rate (0.2 ml/min) by the estimated particle concentration, which depends on the particle class (1.6×10^4 and $0.91 \times 10^4 \, \frac{\text{particles}}{\text{ml}}$ respectively for class A and B) since the mixtures have a common solid content volume. Note that we are assuming that the number of particles that remain stuck somewhere before reaching the illumination area is negligible w.r.t. the total number of passing particles. Therefore, even if we deem this assumption sufficiently true in our case, we should keep in mind that the estimated R_f is more an upper limit for the true particle flow rate. From the next calculation steps it will be evident that this implies that we will obtain a lower limit estimate of the true FoV. To provide an example calculation, assuming a reasonable FoV$= 100\,\mu\text{m}$, respectively for classes A and B we obtain (keeping 2 significant digits): $Pr_A(k = 0)$ = 0.98, $Pr_B(k = 0) = 0.99$, $Pr_A(k = 1)$ = 0.017, $Pr_B(k = 1) = 0.0098$, $Pr_A(k = 2)$ = 0.00016, $Pr_B(k = 2) = 0.000048$. These results are qualitatively consistent with both our visual checks and our assumption that the particles do not significantly often interact during their passage through the microfluidic channel (statistical independence). It should be stressed that, given the low occurence in the training sample set, we do not expect the classifier to learn how to deal with patterns generated when more than one particle is present in the FoV. The particle ratio R can be estimated by $R = 1 - Pr(0, \tau + \text{FoV}/v, R_f)$, with reference to equation (5.4). Thus, by inverting it, we can finally estimate the FoV corresponding to a chosen value of R:

$$\text{FoV} = -\frac{\ln(1-R)v}{R_f} - \tau v \tag{5.5}$$

For each chosen value of R and for each particle class, we report in Table 5.3 the number of classification samples (accepted images) and the FoV estimates. The corresponding estimated FoV is quite large: $\approx$ 0.3 mm. It should also be stressed that, as a consequence of our choice of having a single threshold θ_P for both classes and for training and testing, the FoV was class-dependent.

No diffractive layer

Particle rate	# accepted images		Field of view (mm)	
	class A	class B	class A	class B
0.02	1427	2108	0.09	0.25
0.04	4008	3067	0.27	0.37
0.06	6452	4120	0.45	0.51
0.08	7954	6051	0.56	0.76

Diffraction grating

Particle rate	# accepted images		Field of view (mm)	
	class A	class B	class A	class B
0.02	1416	2288	0.08	0.27
0.04	4173	3213	0.27	0.38
0.06	6826	4207	0.45	0.51
0.08	8354	6199	0.57	0.76

Table 5.3: Correspondence between chosen particle ratio R values (same for both particle classes), the number of images accepted as samples for classification (with strong enough particle signal) and estimated FoV of the classification process. *Left* and *right* tables regard respectively the configurations with and without a diffraction grating interposed between the microfluidic channel and the camera (NDG and DG configurations).

References

[1] Alessio Lugnan, Emmanuel Gooskens, Jeremy Vatin, Joni Dambre, and Peter Bienstman. *Machine learning issues and opportunities in ultrafast particle classification for label-free microflow cytometry.* Scientific Reports, 10(1), 2020.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[3] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. *Shortcut learning in deep neural networks.* Nature Machine Intelligence, 2(11):665–673, 2020.

[4] Ruey-Jen Yang, Lung-Ming Fu, and Hui-Hsiung Hou. *Review and perspectives on microfluidic flow cytometers.* Sensors and Actuators B: Chemical, 266:26–45, 2018.

[5] Shifei Ding, Han Zhao, Yanan Zhang, Xinzheng Xu, and Ru Nie. *Extreme learning machine: algorithm, theory and applications.* Artificial Intelligence Review, 44(1):103–115, 2015.

[6] Weipeng Cao, Xizhao Wang, Zhong Ming, and Jinzhu Gao. *A review on neural networks with random weights.* Neurocomputing, 275:278–287, 2018.

[7] Mantas Lukoševičius and Herbert Jaeger. *Reservoir computing approaches to recurrent neural network training.* Computer Science Review, 3(3):127–149, 2009.

[8] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. *Recent advances in physical reservoir computing: A review.* Neural Networks, 115:100–123, 2019.

[9] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala. *Random projections through multiple optical scattering: Approximating kernels at the speed of light.* In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6215–6219. IEEE, 2016.

[10] Tiankuang Zhou, Xing Lin, Jiamin Wu, Yitong Chen, Hao Xie, Yipeng Li, Jingtao Fan, Huaqiang Wu, Lu Fang, and Qionghai Dai. *Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit.* Nature Photonics, pages 1–7, 2021.

[11] Young Jin Heo, Donghyeon Lee, Junsu Kang, Keondo Lee, and Wan Kyun Chung. *Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip.* Scientific reports, 7(1):1–9, 2017.

[12] Bruno Cornelis, David Blinder, Bart Jansen, Liesbet Lagae, and Peter Schelkens. *Fast and robust Fourier domain-based classification for on-chip lens-free flow cytometry.* Optics Express, 26(11):14329–14339, 2018.

[13] Yueqin Li, Ata Mahjoubfar, Claire Lifan Chen, Kayvan Reza Niazi, Li Pei, and Bahram Jalali. *Deep cytometry: deep learning with real-time inference in cell sorting and flow cytometry.* Scientific reports, 9(1):1–12, 2019.

[14] Keisuke Goda, Ali Ayazi, Daniel R Gossett, Jagannath Sadasivam, Cejo K Lonappan, Elodie Sollier, Ali M Fard, Soojung Claire Hur, Jost Adam, Coleman Murray, et al. *High-throughput single-microparticle imaging flow analyzer.* Proceedings of the National Academy of Sciences, 109(29):11630–11635, 2012.

[15] Queenie TK Lai, Kelvin CM Lee, Anson HL Tang, Kenneth KY Wong, Hayden KH So, and Kevin K Tsia. *High-throughput time-stretch imaging flow cytometry for multi-class classification of phytoplankton.* Optics Express, 24(25):28170–28184, 2016.

[16] Ethan Schonbrun, Sai Siva Gorthi, and Diane Schaak. *Microfabricated multiple field of view imaging flow cytometry.* Lab on a Chip, 12(2):268–273, 2012.

[17] Liesbet Lagae, Dries Vercruysse, Alexandra Dusa, Chengxun Liu, Koen de Wijs, Richard Stahl, Geert Vanmeerbeeck, Bivragh Majeed, Yi Li, and Peter Peumans. *High throughput cell sorter based on lensfree imaging of cells.* In 2015 IEEE International Electron Devices Meeting (IEDM), pages 13–3. IEEE, 2015.

6
Conclusion

In this dissertation we explored a new machine learning approach for versatile and computationally cheap classification of cells or microparticles in label-free imaging flow cytometry. In particular we focused on the direct application of linear classifiers to cell or microparticle interference patterns obtained through the in-line holographic microscopy method, bypassing the image reconstruction step. To make this approach powerful enough for practical implementations, we investigated how we could act on the optical path of the laser beam illuminating the flowing objects, in order to enhance the computational power of linear classification. This technique can be considered a hardware-based version of the extreme learning machine (ELM) method, where a neural network with fixed random internal connections is considered and only the output connections are trained, greatly simplifying the required training operations.

We first investigated such a principle of operation numerically, by means of optical FDTD simulations. In particular, we showed that by interposing microscopic on-chip optical scatterers between simulated flowing cells and a virtual image sensor, we obtained an error reduction of at least 50% in cell classification based on average nucleus size or shape. Interestingly, a similar performance improvement was achieved using many different optical scatterers configurations, both placing the virtual image sensor in the near field or in the far field region. This technique allows for versatile and robust linear classification improvement without any increase in computational cost.

Afterwards, we developed a proof-of-concept imaging flow cytometry experiment, to demonstrate the classification of transparent microparticles on the basis of their average diameter (15.2µm and 18.6µm), through the aforementioned hardware ELM approach. In spite of the high sample variability due to the large field of view of the cytometer (around 0.3mm), we obtained a satisfying classification accuracy (higher than 90%) for different image resolutions, down to 32×26 pixels. We achieved these results at an extremely low computational cost, with an execution time of the whole classification pipeline of 13µs on a common laptop. This is at least one order of magnitude faster than the algorithms reported in other comparable works about fast particle or cell classification, which were accelerated by a GPU.

Furthermore, we experimentally explored the interposition of different types of optical diffractive layers, none of which seemed to improve the classification accuracy. We ascribe this to one (or a combination of) the following possible reasons: the lower signal-to-noise ratio because of the optical attenuation of the diffractive layers; the type of classification task, whose difficulty mostly originates from the large particle displacement w.r.t. the illumination center, instead of from the detection of morphology differences; the limited number of scatterer configurations that was explored. Nevertheless, we demonstrated the robustness of the proposed method, since similar classification accuracy was achieved when the particle patterns were heavily distorted by an interposed diffraction grating.

Properly addressing the shortcut learning issue arising from the influence of slow drifts in measurement conditions on the acquired images, was key to achieve the aforementioned results. We first encountered and investigated this problem, called *measurement bias*, in the attempt to develop a computationally efficient machine learning classification pipeline for the white blood cell holograms provided by our collaborators from imec. Afterwards, through our own particle classification experiment, we demonstrated a methodology to evaluate and treat measurement bias. In particular, we performed several chronologically intertwined measurements of the two particle classes, in order to break the correlation between drifts in measurement conditions and class labels, in the acquired interference patterns. The treatment of shortcut learning, and in particular of measurement bias in imaging flow cytometry, is often not reported in works about new machine learning applications. If left untreated, it might lead to significantly inflated and non generalizable classification performance estimations.

Finally, the all-round simplicity and the low cost of the presented flow cytometry implementation make it suitable for compact point-of-care applications, where both the training and the use of the cytometer should not require high technical expertise.

Perspectives

The approach investigated and developed throughout this dissertation is novel both in its application and, up to a certain depth, in its underlying principles. Therefore, there is plenty of room for further developments along various research directions.

The most straightforward continuation of the presented work is to further explore how the interposition of diffractive layers can enhance the linear classification of the acquired interference patterns. This could be done by employing new optical configurations and/or by considering other classification tasks, e.g. more morphology-based, such as in white blood cell classification. In particular, demonstrating satisfying performances in biological cell classification employing the proposed approach would substantially increase its impact.

Moreover, both form a purely machine learning perspective and application-wise, it would be interesting to directly compare the proposed method with the use of other conventional machine learning algorithms, e.g. based on feature engineering or convolutional neural networks.

An impactful application for our technique would be to integrate it in existing high-throughput imaging flow cytometers, e.g. based on optofluidic time-stretch microscopy or parallel multi-channel operations, in order to enable online analysis for high-speed cell sorting.

Finally, it would be interesting to pass from the spatial ELM approach based on optical hardware to its temporal counterpart, exploiting the dynamics of the projected interference pattern due to the particle movement. An interposed diffractive layer could be employed to implement random input connections of the corresponding neural network scheme. In practice, the image sensor could be substituted by a high-speed photodiode, so as to switch to single-pixel detection. This would remove the heavy throughput constraints due to the camera frame rate. The light projected by the cell passage through the diffractive layer to the photodiode would be recorded as a time-dependent perturbation. The corresponding time series can then be employed as samples to train a linear classifier. Assuming that sufficiently high classification performance could be achieved, this technique would directly enable high-throughput online classification with relatively cheap and simple components.